XMIT-Funktionsblock
Benutzerhandbuch

840USE11302 Version 4.0

31005379 00

eine Marke von
Schneider
Electric

Telemecanique

Inhaltsverzeichnis

Sicherheitshinweise 7

Über dieses Buch 9

Kapitel 1 Einführung in XMIT 11
Auf einen Blick 11
XMIT-spezifische Funktionalität 12
Produkte von Schneider, die XMIT unterstützen 14
Kenndaten des XMIT-Loadable 16
PLC-Loadable-Funktionen 18
Kenndaten der eingebauten XMIT-Funktion 19
Kundendienst 20

Kapitel 2 Installation des Funktionsblocks für das XMIT-Loadable . . . 21
Auf einen Blick 21
2.1 Installation des XMIT-Loadable mit Modsoft 22
Auf einen Blick 22
Übertragen des Loadable auf eine SPS und Auswahl der Optionen mit Hilfe von Modsoft 23
Laden der Modsoft XMIT-Zoom- und -Hilfefenster 25
Laden der Datei NSUP.EXE mit Hilfe von Modsoft 27
Laden der Datei XMIT.EXE mit Hilfe von Modsoft 29
2.2 Installieren des XMIT-Loadable mit Concept und ProWORX 32
Auf einen Blick 32
Laden der NSUP- und XMIT-Loadables unter Verwendung von Concept 33
Laden der NSUP- und XMIT-Loadables unter Verwendung von ProWORX NxT 36
Laden der NSUP- und XMIT-Loadables unter Verwendung von Using ProWORX32 41

Kapitel 3 Verwendung von Zoomfenstern 49
Auf einen Blick 49
Zoomfenster für den Kommunikationsblock in Modsoft 50
Zoomfenster für den Schnittstellenstatusblock in Modsoft 54
Zoomblöcke für Konvertierungsblocks in Modsoft 56

Zoomfenster, in denen Concept verwendet wird 58
Zoomfenster, in denen ProWORX NxT verwendet wird 62
Zoomfenster, in denen ProWORX32 verwendet wird 65

Kapitel 4 Verwendung des XMIT-Funktionsblocks 69
Auf einen Blick 69
4.1 Beschreibung des XMIT-Kommunikationsblocks 70
Auf einen Blick 70
XMIT- und SPS-Kompatibilität 71
Struktur des XMIT-Funktionsblocks 73
Inhalt des XMIT-Eintrags 75
XMIT-Kommunikationsfunktionen 77
4.2 Verwendung der XMIT-Kommunikationsblockregister 78
Auf einen Blick 78
XMIT-Kommunikationsblock-Register 4x bis 4x + 7 79
XMIT-Kommunikationsblock, Register 4x + 8 82
XMIT-Kommunikationsblock-Register 4x + 8, Bit 5 86
XMIT-Kommunikationsblock-Register 4x + 8, Bit 6 89
XMIT-Kommunikationsblock-Register 4x + 8, Bit 9 90
XMIT-Kommunikationsblock-Register 4x + 8, Bit 10 91
XMIT-Kommunikationsblock-Register 4x + 8, Bit 11 92
XMIT-Kommunikationsblock-Register 4x + 8, Bit 12 94
XMIT-Kommunikationsblock-Register 4x + 9, Funktionscodes 01 bis 06, 15 und 16 96
XMIT-Kommunikationsblock-Register 4x + 9, Funktionscode 8 98
XMIT-Kommunikationsblock-Register 4x + 9, Funktionscodes 20 und 21 100
XMIT-Kommunikationsblock-Register 4x + 10 bis 4x + 15 102
4.3 Beschreibung und Verwendung des XMIT-Schnittstellenstatusblocks 104
Auf einen Blick 104
XMIT-Schnittstellenstatusblock und SPS-Kompatibilität 105
Darstellung des XMIT-Funktionsblocks für den Schnittstellenstatus und Inhalt des oberen Eintrags 107
Tabelle der XMIT-Schnittstellenstatusanzeigen 109
4.4 Beschreibung des XMIT-Konvertierungsblocks 112
Auf einen Blick 112
XMIT-Konvertierungsblock und SPS-Kompatibilität 113
Struktur und Inhalt des XMIT-Konvertierungsblocks 115
4.5 Verwendung des XMIT-Konvertierungsblocks 117
Auf einen Blick 117
XMIT-Konvertierungsblock-Register 4x bis 4x + 2 118
XMIT-Konvertierungsblock, Register 4x + 3 119
XMIT-Konvertierungsblock-Register 4x + 4 bis 4x + 7 121
4.6 Beispiele für den XMIT-Konvertierungsblock-Opcode 123
Auf einen Blick 123
XMIT-Konvertierungs-Opcode – Beispiele 1 bis 3 124
XMIT-Konvertierungsblock-Opcode – Beispiele 4 bis 6 127

XMIT-Konvertierungsblock-Opcode – Beispiele 7 bis 11 129
XMIT Binär/BCD-Konvertierungstypen 132

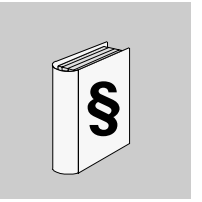
Kapitel 5 Arbeiten mit XMIT-Beispielen 133
Auf einen Blick. 133
5.1 Einfache ASCII-Lese-/-Schreibvorgänge und Modbus-Lese-/-Schreibvorgänge 134
Auf einen Blick. 134
Lesen von einfachen ASCII-Zeichen mit Hilfe von Concept 135
Schreiben von einfachen ASCII-Zeichen mit Hilfe von ProWORX32 139
Modbus-Lesevorgänge, bei denen ProWORX NxT verwendet wird 143
Modbus-Schreibvorgänge an RS485-Schnittstelle Nr. 2 unter Verwendung von ProWORX NxT 147
5.2 Übertragen mehrerer Modbus-Befehle: SPS-Master an SPS-Slave 151
Auf einen Blick. 151
Senden mehrerer Modbus-Befehle 152
Einrichten der Master-SPS 153
Verwendung der Ladder Logic für mehrere Modbus-Befehle – Netzwerk Nr. 1. 154
Verwendung der Ladder Logic für mehrere Modbus-Befehle – Netzwerk Nr. 2. 155
Verwendung der Ladder Logic für mehrere Modbus-Befehle – Netzwerk Nr. 3. 158
Verwendung der Ladder Logic für mehrere Modbus-Befehle – Netzwerk Nr. 4. 160
Abschließende Übertragung mehrerer Modbus-Befehle. 161
5.3 Übertragung des Fehlerworts über Impulswahlmodems an den SPS-Slave . . 162
Auf einen Blick. 162
Übertragung des Fehlerworts an die Slave-SPS über Impulswahlmodems . . . 163
Modem-Setup 164
Einrichten der Master-SPS 165
Verwendung der Ladder Logic für die Fehlerwortübertragung – Netzwerk Nr. 1. 166
Verwendung der Ladder Logic für die Fehlerwortübertragung – Netzwerk Nr. 2. 167
Verwendung der Ladder Logic für die Fehlerwortübertragung – Netzwerk Nr. 3. 170
Verwendung der Ladder Logic für die Fehlerwortübertragung – Netzwerk Nr. 4. 172
Abschließende Übertragung des Fehlerworts. 173

Anhang 175
Auf einen Blick 175

Anhang A Technische Angaben zu XMIT 177
Auf einen Blick 177
A.1 Arbeiten mit Anfrage-/Antwortparametern 178
Grenzwerte für Modbus-Anfrage-/Antwortparameter 178
A.2 Kabelbelegung und Adapter 181
Auf einen Blick 181
Kabelbelegungen 182
9-Pin auf 25-Pin (Modem) OHNE RTS/CTS-Datenflusssteuerung 184
9-Pin (RS-232) auf 25-Pin (Modem) mit RTS/CTS-Datenflusssteuerung 185
9-Pin auf 25-Pin (Nullmodem) 186
9-Pin auf 9-Pin (Modem) 187
9-Pin auf 9-Pin (Nullmodem) 188
RJ-45-(8x8) auf 25-poligen Stecker (Modem) (Konfiguration A) 189
RJ-45-(8x8) auf 25-poligen Stecker (Modem) (Konfiguration B) 190
RJ-45-(8x8) auf 25-Pin-Stecker (Nullmodem) 191
RJ-45-(8x8) 9-Pin-Stecker (Modem) (Konfiguration A) 192
RJ-45-(8x8) 9-Pin-Stecker (Modem) (Konfiguration B) 193
RJ-45 (8x8) 9-poliger Stecker (Nullmodem) 194
RJ-45-(8x8) auf RJ-45-(8x8) (Modem) 195
Teilesätze für Kabeladapter 196
A.3 Konfiguration von XMIT ausschließlich mit Hayes-kompatiblen Impulswahlmodems 197
Auf einen Blick 197
Verwendung der XMIT-Konfiguration ausschließlich mit Hayes-kompatiblen Impulswahlmodems 198
Verwendung von Initialisierungsnachrichten mit Hayes-Modems 199
Verwendung von Impulswahlmodem-Meldungen mit Hayes-Modems 201
Verwendung von Auflegenachrichten ausschließlich mit Hayes-kompatiblen Impulswahlmodems 202

Index 203

Sicherheitshinweise



Wichtige Informationen

HINWEIS

Lesen Sie diese Anweisungen gründlich durch und machen Sie sich mit dem Gerät vertraut, bevor Sie es installieren, in Betrieb nehmen oder warten. Die folgenden Hinweise können an verschiedenen Stellen in dieser Dokumentation enthalten oder auf dem Gerät zu lesen sein. Die Hinweise warnen vor möglichen Gefahren oder machen auf Informationen aufmerksam, die Vorgänge erläutern bzw. vereinfachen.

Erscheint dieses Symbol zusätzlich zu einem Warnaufkleber, bedeutet dies, dass die Gefahr eines elektrischen Schlags besteht und die Nichtbeachtung des Hinweises Verletzungen zur Folge haben kann.

Dies ist ein allgemeines Warnsymbol. Es macht Sie auf mögliche Verletzungsgefahren aufmerksam. Beachten Sie alle unter diesem Symbol aufgeführten Hinweise, um Verletzungen oder Unfälle mit Todesfolge zu vermeiden.

GEFAHR

GEFAHR macht auf eine unmittelbar gefährliche Situation aufmerksam, die bei Nichtbeachtung **unweigerlich** einen schweren oder tödlichen Unfall oder Beschädigungen an Geräten zur Folge hat.

WARNUNG

WARNUNG macht auf eine möglicherweise gefährliche Situation aufmerksam, die bei Nichtbeachtung **unter Umständen** einen schweren oder tödlichen Unfall oder Beschädigungen an Geräten zur Folge hat.

VORSICHT

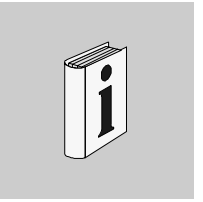
VORSICHT macht auf eine möglicherweise gefährliche Situation aufmerksam, die bei Nichtbeachtung **unter Umständen** einen Unfall oder Beschädigungen an Geräten zur Folge hat.

BITTE BEACHTEN

Elektrische Geräte dürfen nur von Fachpersonal gewartet und instandgesetzt werden. Schneider Electric haftet nicht für Schäden, die aufgrund der Verwendung dieses Materials entstehen. Dieses Dokument ist nicht als Betriebsanleitung für nicht geschultes Personal vorgesehen.

© 2003 Schneider Electric. Alle Rechte vorbehalten.

Über dieses Buch



Auf einen Blick

Ziel dieses Dokuments

Dieses Benutzerhandbuch enthält alle erforderlichen Informationen für die ladbare und die integrierte Version des (984LL) XMIT-Funktionsblocks, der auf allen SPS-Hardware-Plattformen zusammen mit der Steuerungs-Software Modsoft, Concept (alle Versionen), ProWORX NxT (alle Versionen) oder ProWORX32 (alle Versionen) verwendet werden kann.

Hinweis: In diesem Handbuch bezieht sich der Ausdruck "Steuerungs-Software" auf Modsoft, Concept, ProWORX NxT oder ProWORX32.

Hinweis: Dieses Benutzerhandbuch enthält KEINE Informationen zum IEC XXMIT-Funktionsblock, der mit der Steuerungs-Software Concept verwendet wird.

Gültigkeitsbereich

Die Daten und Abbildungen in diesem Buch sind nicht verbindlich. Wir behalten uns das Recht vor, unsere Produkte gemäß unseren Richtlinien zur ständigen Produktverbesserung zu ändern. Die in diesem Dokument enthaltenen Informationen können jederzeit unangekündigt geändert werden und stellen keine verbindlichen Zusagen seitens Schneider Electric dar.

Weiterführende Dokumentation

Titel	Referenz-Nummer
Referenzhandbuch für Modicon Modbus-Protokolle	PI-MBUS-300
Benutzerhandbuch für die 984-A120 Compact-SPS	890USE10802
Modicon Quantum-Automationsserie – Hardware-Referenzhandbuch	840USE10002
Momentum M1 – CPU- und Schnittstellen-Adapter – Benutzerhandbuch	870USE10112
Modicon 512/612 Micro-SPS-Hardware – Benutzerhandbuch	890USE14502
Modicon Micro-Controller-Ladder Logic – Benutzerhandbuch	890USE14602
Modicon Modsoft Benutzerhandbuch für Programmierer	890USE11502
Benutzerhandbuch zur Modicon-KOP-Bausteinbibliothek	840USE10102
Modicon 309COM4550x XMIT-Loadable – Readme	GI-XMIT-RMF

Produkt-bezogene Warnhinweise

Bei der Installation und der Verwendung dieses Produkts müssen alle relevanten staatlichen, regionalen und lokalen Sicherheitsbestimmungen eingehalten werden. Aus Sicherheitsgründen und um die Übereinstimmung mit dokumentierten Systemdaten zu gewährleisten, dürfen Reparaturen an den Komponenten nur durch den Hersteller durchgeführt werden.

Schneider Electric übernimmt keine Verantwortung für etwaige in diesem Dokument enthaltene Fehler. Wenn Sie Verbesserungs- oder Ergänzungsvorschläge haben oder Fehler in dieser Veröffentlichung gefunden haben, bitte benachrichtigen Sie uns.

Kein Teil dieses Dokuments darf ohne die ausdrückliche schriftliche Erlaubnis durch den Herausgeber Schneider Electric in irgendeiner Weise oder mit irgendeinem Verfahren elektronischer oder mechanischer Art, auch nicht durch Fotokopieren, vervielfältigt werden.

Benutzer-kommentar

Ihre Anmerkungen und Hinweise sind uns jederzeit willkommen. Senden Sie sie einfach an unsere E-mail-Adresse: TECHCOMM@modicon.com

Einführung in XMIT

1

Auf einen Blick

Einführung

In diesen Unterlagen erhalten Sie Informationen zum XMIT-Funktionsblock.

Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Thema	Seite
XMIT-spezifische Funktionalität	12
Produkte von Schneider, die XMIT unterstützen	14
Kenndaten des XMIT-Loadable	16
PLC-Loadable-Funktionen	18
Kenndaten der eingebauten XMIT-Funktion	19
Kundendienst	20

XMIT-spezifische Funktionalität

XMIT-Beschreibung

Der Funktionsblock für die Übertragung (XMIT), der entweder als ladbares Hauptsteuerprogramm oder als integrierte Funktion verfügbar ist, bietet folgende Möglichkeiten:

- ASCII-Meldungen
  - Einfaches ASCII
  - Abgeschlossenes ASCII
- Modbus-Meldungen
- Statusinformationen zur Schnittstelle
- Datenkonvertierung

XMIT-Verfügbarkeit

Der XMIT-Funktionsblock kann geladen werden auf

- Quantum
- Compact
- Micro

Der XMIT-Funktionsblock ist integriert in

- Momentum

XMIT-Modi

XMIT-Modi:

- Kommunikationsmodus
- Modus für Schnittstellenstatus
- Konvertierungsmodus

XMIT-Eingänge

Sie können (1) ASCII- oder binäre Daten in Ihre SPS importieren bzw. daraus exportieren oder (2) Daten entsprechend den Anforderungen Ihrer Anwendung in verschiedene binäre Formate oder ASCII konvertieren, um sie an DCE-Geräte zu senden. Detaillierte Informationen zu XMIT-Kommunikationsblöcken finden Sie unter *Verwendung der XMIT-Kommunikationsblockregister, S. 78*, detaillierte Informationen zum XMIT-Schnittstellenstatusblock erhalten Sie unter *Beschreibung und Verwendung des XMIT-Schnittstellenstatusblocks, S. 104* und Informationen zum XMIT-Konvertierungsblock finden Sie unter *Verwendung des XMIT-Konvertierungsblocks, S. 117*.

XMIT-Diagnose

Der XMIT-Funktionsblock verfügt über eingebaute Diagnostikfunktionen, mit denen sichergestellt wird, dass auf der SPS keine anderen XMIT-Blöcke aktiv sind. Innerhalb des XMIT-Funktionsblocks können Sie mit einer Steuertabelle die Kommunikationsverbindung zwischen der SPS und dem DCE (Data Communication Equipment, Datenkommunikationsgerät) steuern, das an Modbus-Schnittstelle Nr. 1 oder 2 der SPS angeschlossen ist. Während der Datenübertragung wird die Schnittstellen-LED NICHT durch den XMIT-Block aktiviert.

Hinweis: Vermeidung von konkurrierenden Zugriffen und Datenkollisionen

Beim Modbus-Protokoll handelt es sich um ein Protokoll mit einem Master und mehreren Slaves.

Für Modbus ist die Verwendung von einem einzelnen Master erforderlich, der mehrere Slaves abfragt. Wenn Sie daher den XMIT-Funktionsblock in einem Netzwerk mit mehreren Mastern verwenden, müssen Sie sich um die Vermeidung konkurrierender Zugriffe und Datenkollisionen kümmern. Die Vermeidung von konkurrierenden Zugriffen und Datenkollisionen lässt sich auf einfachem Wege über Benutzerlogik-Programmierung erreichen.

XMIT- und Modbus-Meldungen

Der XMIT-Funktionsblock sendet entweder (1) Modbus-Meldungen von einer Master-SPS an mehrere Slave-SPS oder (2) ASCII-Zeichenketten von Modbus-Slave-Schnittstelle 1 oder 2 der SPS an ASCII-Drucker und -Terminals. Der XMIT-Funktionsblock sendet diese Meldungen über Telefonwähl-Modems, Radio-Modems oder einfach über eine Direktverbindung.

Produkte von Schneider, die XMIT unterstützen

Einschränkungen der XMIT-Funktionalität

Produkte von Schneider, die den XMIT-Funktionsblock unterstützen

XMIT-Funktionsblock	Produktfamilie	Modellnummern
Steuert Schnittstelle Nr. 1 bzw. Nr. 2 auf	Momentum	171CCS70000
		171CCS70010
		171CCS76000
		171CCC76010
		171CCS78000
		171CCC78010
Steuert Schnittstelle Nr. 1 auf	Quantum	140CPU11302
		140CPU11303
		140CPU21304
		140CPU42402
		140CPU43412
		140CPU53412
		140CPU43412A
		140CPU53414A
	Compact	PC E984 2xx SPS
		984-E258
		984-E265
		984-E275
		984-E285
Steuert Schnittstelle Nr. 2 auf	Micro	110CPU61204
	Momentum	171CCC98020
		171CCC96020
		171CCC98030
		171CCC96030
		171CCC98091
		171CCC96091
NICHT betriebsbereit auf	Micro	110CPU512 xx
		110CPU61200/03
	Compact	PC A984 1xx
		PC 0984 1xx

Für Modbus-Abfrage-/Antwortparameter gelten Grenzwerte, die vom jeweiligen SPS-Modell abhängen.
(Siehe *Grenzwerte für Modbus-Anfrage-/Antwortparameter, S. 178*.)

Kenndaten des XMIT-Loadable

Funktionsblock des XMIT-Loadable

Die folgenden Informationen gelten für den Funktionsblock des XMIT-Loadable.

- Modsoft Version 2.5 oder niedriger (Teilenummer SW MSxD 9SA)
- Modsoft Version 2.6 oder höher (Teilenummer SW MSxD 9SA)
- Funktionsblock des XMIT-Loadable (Teilenummer 309 COM 455 0x)
  Enthält vier Dateien und zwei Unterverzeichnisse:
  - README.TXT
  - NSUP.EXE
  - XMIT1968.HLP
  - XMIT.EXE
  - /MS_25
    Enthält die Datei DXFDT.SYS
  - /MS_26
    Enthält die Datei XMIT.ZMM

Nicht-PC-Ausführungs-programm

Die Loadable-Version des XMIT-Blocks ist für die folgenden Modelle verfügbar:

Produktfamilie	Modellnummer
Quantum	140CPU11302
	140CPU21304
	140CPU11303 (mit Hauptsteuerprogramm 2.12 oder höher)
	140CPU42402 (mit Hauptsteuerprogramm 2.10 oder höher)
	140CPU43412
	140CPU53414 (mit Hauptsteuerprogramm 1.02 oder höher)
	140CPU43412A
	140CPU53414A
Compact	PC E984 241
	PC E984 245
	PC E984 251
	PC E984 255 (mit Hauptsteuerprogramm 1.02 oder höher)
	984-E258
	984-E265
	984-E275
	984-E285
Micro	110 CPU 612 04 (mit Hauptsteuerprogramm 1.00 oder höher)
Kommunikationsmedien	Impulswahlmodems
	Mietleitungs-Modems
	Weitere Informationen über Kommunikationsmedien finden Sie in der Liste der getesteten Modems und Drucker im Dokument Modicon 309COM4550x XMIT-Loadable – Readme (GI-XMIT-RMF).

PLC-Loadable-Funktionen

Übersicht der Loadables

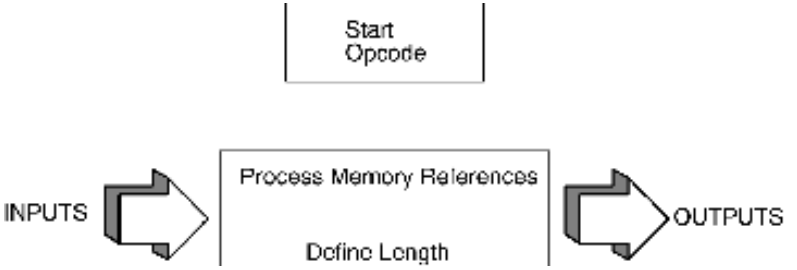
In einer SPS sind Tabellen nit Konfigurations-Daten vorhanden. Ladbare Funktionsblöcke, ein in der Anwendung programmierter ausführbarer Software-Code, können der Steuerung bzw. der SPS hinzugefügt werden.

Für diese Loadable-Funktionen gilt Folgendes

- Sie sind anwendungsspezifische programmierbare Blöcke, die in der SPS geladen werden
- Sie ermöglichen die Einstellung von Opcodes über die Steuerungs-Software
- "Konfigurieren" Sie die Loadable-Funktionen im größeren Steuerungsprogramm

Der ausführbare Software-Code wird in der Anwendung im Format eines standardmäßigen KOP-Anweisungs-Blocks mit drei Einträgen programmiert. Die grundlegende Logik-Unterprogramm-Struktur eines Loadable wird in der folgenden Abbildung dargestellt.

Der logische Fluss des Loadable-Code



In einer Feldumgebung können Loadable-Funktionen der vorhandenen Steuerungs-Logik hinzugefügt werden und bieten eine Software-Lösung für spezielle Anwendungsprobleme.

Kenndaten der eingebauten XMIT-Funktion

Eingebauter XMIT-Funktionsblock

Erforderlich

- Modsoft Version 2.6 oder höher (Teilenummer SW MSxD 9SA).

Die integrierte Version des XMIT-Blocks ist für die folgenden Modelle verfügbar:

Produktfamilie	**Modellnummer**
Momentum	171CCS70000
	171CCS70010
	171CCS76000
	171CCS78000
	171CCC76010
	171CCC78010 (mit Hauptsteuerprogramm 2.00 oder höher)
	171CCC98020
	171CCC96020
	171CCC98030
	171CCC96030
	171CCC98091
	171CCC96091
Momentum-Steuerungen unterstützen nur ein Stoppbit.	
Kommunikationsmedien	Impulswahlmodems
	Mietleitungsmodem
	Weitere Informationen über Kommunikationsmedien finden Sie in der Liste der getesteten Modems und Drucker im Dokument Modicon 309COM4550x XMIT-Loadable – Readme (GI-XMIT-RMF).

Kundendienst

Kontakt-informationen Kundendienst

Die Telefonnummern von Schneider Automation lauten wie folgt:

- Für Anrufer aus Nordamerika (außer Massachusetts):
  (800) 468 5342
- Für Anrufer aus Massachusetts:
  (978) 975 5001
- Für Anrufer, die nicht aus Nordamerika anrufen:
  1 (978) 975 5001
- Fax:
  (978) 975 9301

Installation des Funktionsblocks für das XMIT-Loadable

2

Auf einen Blick

Überblick

In diesem Kapitel wird die Installation des Funktionsblocks für das XMIT-Loadable beschrieben. Darüber hinaus erhalten Sie wichtige Informationen zu den Dateien, die in diesem Block enthalten sind.

Vor dem Laden sollten Sie sich mit der Steuerungs-Software Modsoft vertraut gemacht, die SPS konfiguriert haben und bereit sein zum Laden von XMIT.

Die Abbildungen in diesem Abschnitt dienen als Beispiele für die Bildschirme, die Sie bei der Übertragung des Loadable XMIT von der Festplatte auf die 984-Steuerung und bei der Verwendung von XMIT angezeigt bekommen.

Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Abschnitte:

Abschnitt	Thema	Seite
2.1	Installation des XMIT-Loadable mit Modsoft	22
2.2	Installieren des XMIT-Loadable mit Concept und ProWORX	32

2.1 Installation des XMIT-Loadable mit Modsoft

Auf einen Blick

Zweck

In diesem Abschnitt wird die Installation des Funktionsblocks für das XMIT-Loadable unter Verwendung von Modsoft beschrieben. Darüber hinaus erhalten Sie wichtige Informationen zu den Dateien, die in diesem Block enthalten sind.
Vor dem Laden sollten Sie sich mit Modsoft vertraut gemacht, die SPS konfiguriert haben und bereit sein zum Laden von XMIT.
Die Abbildungen in diesem Abschnitt dienen als Beispiele für die Bildschirme, die Sie bei der Übertragung des Loadable XMIT von der Festplatte auf die 984-Steuerung und bei der Verwendung von XMIT angezeigt bekommen.

Inhalt dieses Abschnitts

Dieser Abschnitt enthält die folgenden Themen:

Thema	Seite
Übertragen des Loadable auf eine SPS und Auswahl der Optionen mit Hilfe von Modsoft	23
Laden der Modsoft XMIT-Zoom- und -Hilfefenster	25
Laden der Datei NSUP.EXE mit Hilfe von Modsoft	27
Laden der Datei XMIT.EXE mit Hilfe von Modsoft	29

Übertragen des Loadable auf eine SPS und Auswahl der Optionen mit Hilfe von Modsoft

Vor dem Laden

Sie sollten mit Modsoft vertraut sein, die SPS konfiguriert haben und bereit sein zum Laden von XMIT.
Wenn Sie die Übertragung an die Steuerung abgeschlossen haben, wird DX – einschließlich Konfiguration – auf die Steuerung heruntergeladen.

Übertragen von XMIT von der Festplatte auf eine 984-SPS

Wenn das Loadable auf die Steuerung übertragen wird, wird die Datei `XMIT.EXE` durch Modsoft in eine DX-Datei mit dem Namen `XMIT1968.EXE` konvertiert.

Schritt	Aktion
1	Legen Sie die **Diskette für den XMIT-Loadable-Funktionsblock** (Teilenummer 309 COM 455 0x) in Laufwerk A: ein.
2	Wählen Sie **Offline** (F2) im **Hauptmenü**.
3	Wählen Sie im Menü entweder **Programm auswählen** oder **Neues Programm** aus.
4	Wählen Sie im Menü **Konfiguration** (F5) aus.

Info zu NSUP und LSUP

Hinweis: Die Loadables NSUP und LSUP werden verwendet, um die Loadables `(.EXE-Dateien)` auf das Betriebssystem der SPS zu übertragen.

- Der XMIT-Loadable-Funktionsblock auf der Diskette umfasst vier Dateien und zwei Unterverzeichnisse:
  - `README.TXT`
  - `NSUP.EXE`
  - `XMIT1968.HLP`
  - `XMIT.EXE`
  - `/MS_25`
    Enthält die Datei `DXFDT.SYS`
  - `/MS_26`
    Enthält die Datei `XMIT.ZMM`

Auswahl eines Segments und eines Netzwerks

Wenn Sie ein Segment und ein Netzwerk auswählen möchten, gehen Sie wie folgt vor:

Schritt	Aktion
1	Drücken Sie zweimal die Taste **Esc**.
2	Die **Segment-Statusanzeige** wird angezeigt.
3	Wählen Sie **Element** (F3).
4	Wählen Sie ein Segment und ein Netzwerk aus.
5	Drücken Sie die **Eingabetaste**.

Auswahl eines XMIT-Loadable

Gehen Sie wie folgt vor, um ein spezielles XMIT-Loadable auszuwählen:

Schritt	Aktion
1	Wählen Sie **Loadable** (F5).
2	Wählen Sie in der Liste das erforderliche XMIT-Loadable aus.

Auswahl der Zoomfenster

Für den XMIT-Block stehen vierzehn Zoomfenster zur Verfügung. (Siehe *Zoomfenster für den Kommunikationsblock in Modsoft, S. 50*).

Schritt	Aktion
1	Platzieren Sie den Cursor auf dem XMIT-Block.
2	Drücken Sie **Alt + Z**, um die XMIT-Zoomfenster anzuzeigen.
3	Wählen Sie die Parameter in den Zoomfenstern aus. Wählen Sie die Parameter aus, die für Ihre Anwendung geeignet sind. Im Kapitel *Verwendung des XMIT-Funktionsblocks, S. 69* wird die Einstellung der Systemparameter beschrieben.

Laden der Modsoft XMIT-Zoom- und -Hilfefenster

Laden der DX-Zoomfenster: DXFDT.SYS

Wenn Sie für die DX-Zoomfenster **Modsoft 2.5 (oder niedriger)** verwenden, laden Sie die Datei DXFDT.SYS.

Schritt	Aktion
1	Bitte beachten Sie vor dem Laden Folgendes: • Stellen Sie sicher, dass Sie **Modsoft 2.5 (oder niedriger)** verwenden, um die DX-Zoomfenster verwenden zu können, die mit Hilfe der Datei DXFDT.SYS geladen werden. • Wenn Sie Modsoft 2.5 (oder niedriger) nicht verwenden, erhalten Sie weitere Informationen unter **Laden der DX-Zoomfenster: XMIT.ZMM**.
2	Die Datei DXFDT.SYS wird im Unterverzeichnis /MS_25 gespeichert.
3	Kopieren Sie die Datei DXFDT.SYS in das Verzeichnis Modsoft/runtime.
4	**Hinweis:** Die Datei DXFDT.SYS ersetzt die bereits vorhandene Datei DXFDT.SYS.

Laden der DX-Zoomfenster: XMIT.ZMM

Wenn Sie **Modsoft 2.6 (oder höher)** für die DX-Zoomfenster verwenden, laden Sie die Datei XMIT.ZMM.

Schritt	Aktion
1	Vor dem Laden • Stellen Sie sicher, dass Sie **Modsoft 2.6 (oder höher)** verwenden, um die DX-Zoomfenster verwenden zu können, die mit der Datei XMIT.ZMM geladen werden. • Wenn Sie NICHT Modsoft 2.6 oder höher verwenden, lesen Sie nach unter **Laden der DX-Zoomfenster: DXFDT.SYS**.
2	Die Datei XMIT.ZMM wird im Unterverzeichnis /MS_26 gespeichert.
3	Kopieren Sie die Datei XMIT.ZMM in das Verzeichnis, in dem sich die Programmdateien befinden.
4	**Hinweis:** VERFÜGBARKEIT DER DX-ZOOMFENSTER Die Datei XMIT.ZMM MUSS sich im selben Verzeichnis befinden, in dem die Programmdateien für das Programm gespeichert sind, das XMIT verwendet. Anderenfalls sind die DX-Zoomfenster nicht verfügbar.

Laden der Hilfefenster: XMIT1968.HLP

Führen Sie die folgenden Schritte durch:

Schritt	Aktion
1	Kopieren Sie die Datei `XMIT1968.HLP` in das Verzeichnis, in dem sich die Programmdateien befinden.
2	Hinweis: VERFÜGBARKEIT DES HILFEFENSTERS Diese Datei MUSS sich im selben Verzeichnis befinden wie die Programmdateien des Programms, das XMIT verwendet. Anderenfalls ist das Hilfefenster nicht verfügbar.
3	Wählen Sie **Alt + H**, um das Hilfefenster für XMIT aufzurufen.

Laden der Datei NSUP.EXE mit Hilfe von Modsoft

Vor dem Laden

Hinweis:

- Um den XMIT-Block auf der SPS auszuführen, laden Sie entweder die Datei NSUP.EXE oder die Datei LSUP.EXE.
- Wenn Sie die Datei LSUP.EXE geladen haben, ist ein Laden der Datei NSUP.EXE nicht erforderlich.
- Ladereihenfolge

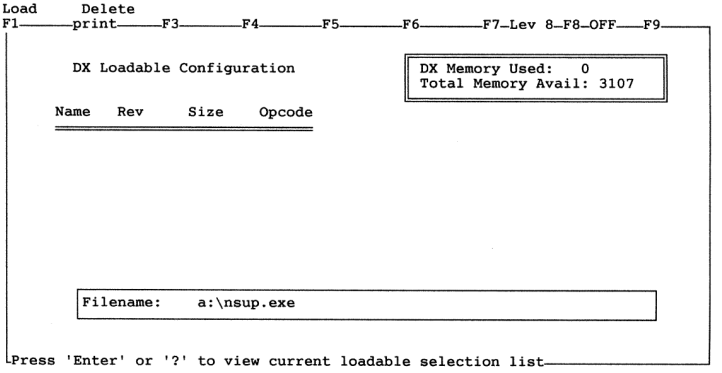
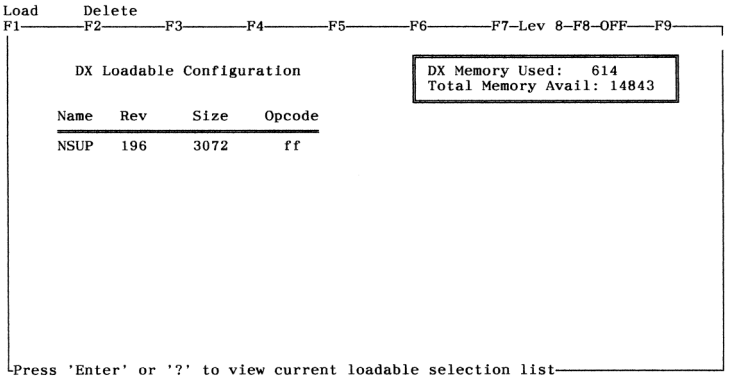
  Sie MÜSSEN die Datei NSUP.EXE laden, BEVOR die Datei XMIT.EXE auf der SPS geladen wird.

  Wenn die Datei XMIT.EXE zuerst geladen wird, funktioniert die XMIT-Anweisung nicht ordnungsgemäß. Dies führt dazu, dass alle drei Ausgänge eingeschaltet werden.
- Wenden Sie sich an den Kundendienst, um die aktuellste Version der Datei NSUP.EXE zu erhalten.

Laden von NSUP.EXE

Führen Sie die folgenden Schritte aus:

Schritt	Aktion
1	Wählen Sie **Loadable** (F5).
2	Wählen Sie **Dir** (F3).
3	Wählen Sie **Laden** (F1). An einer Eingabeaufforderung werden Sie nach dem Dateinamen gefragt.
4	Geben Sie **A:\ NSUP.EXE** ein.

Schritt	Aktion
5	Drücken Sie die **Eingabetaste**. Eine Systemmeldung wird angezeigt, in der Sie darauf hingewiesen werden, dass Sie nun auf dieses Loadable zugreifen können. 
6	Drücken Sie **Umschalt + ?**, um alle verfügbaren Loadables anzuzeigen. Das Loadable `NSUP.EXE` wird in der Liste angezeigt.
7	Platzieren Sie den Cursor auf **NSUP.EXE**.
8	Drücken Sie die **Eingabetaste**. Im Bildschirm werden die Versionsnummer, die Dateigröße und der Opcode für das NSUP-Loadable angezeigt.  Der Opcode des Loadable lautet **ff** (Hex).
9	Hinweis: Stellen Sie sicher, dass dieser Opcode keinen Konflikt mit anderen Opcodes verursacht, die eventuell verwendet werden. Wenn ein Konflikt auftritt, wählen Sie einen neuen Opcode in der Liste aus.

Laden der Datei XMIT.EXE mit Hilfe von Modsoft

Vor dem Laden

Hinweis:

Folgendes geschieht, wenn das NSUP-Loadable

1. NICHT installiert ist oder
2. nach dem XMIT-Loadable installiert wurde oder
3. auf einer Quantum-SPS mit einer älteren Hauptsteuerprogramm-Version als in der entsprechenden Einheit angegeben installiert wurde: *SPS-Kompatibilität, S. 71*

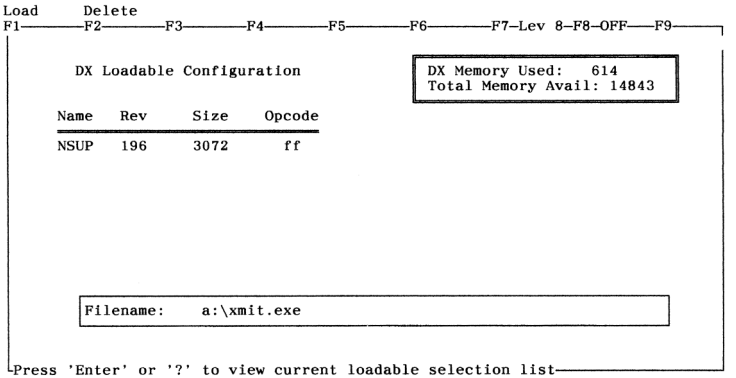
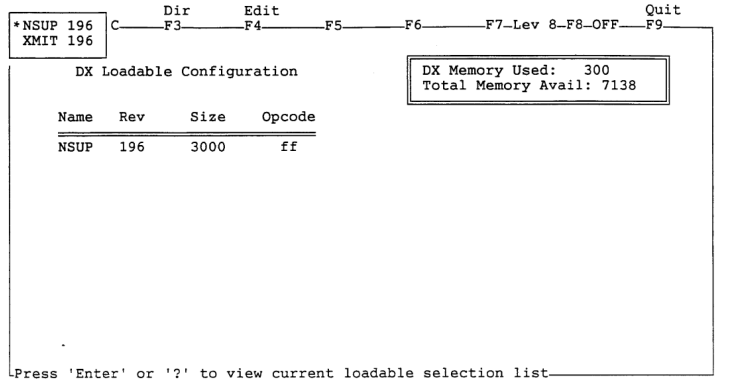
Alle drei Ausgänge werden eingeschaltet, unabhängig von den Eingangszuständen.

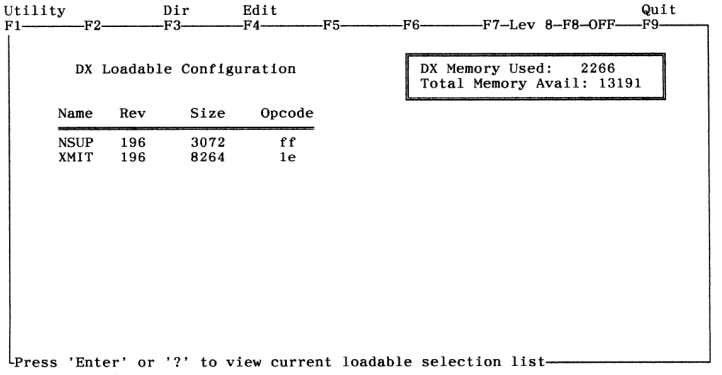
Hinweis: Die Datei `NSUP.EXE` MUSS in der SPS geladen werden, BEVOR die Datei `XMIT.EXE` geladen wird. Ist dies nicht der Fall, ist die XMIT-Anweisung nicht funktionsfähig.

Laden von XMIT.EXE

Führen Sie die folgenden Schritte aus:

Schritt	Aktion
1	Wählen Sie **Loadable** (F5).
2	Wählen Sie **Dir** (F3).
3	Wählen Sie **Laden** (F1). An einer Eingabeaufforderung werden Sie nach dem Dateinamen gefragt.
4	Geben Sie **A:\ XMIT.EXE** ein.

Schritt	Aktion
5	Drücken Sie die **Eingabetaste**. Eine Systemmeldung wird angezeigt, in der Sie darauf hingewiesen werden, dass Sie nun auf das Loadable XMIT zugreifen können. 
6	Bewegen Sie den Cursor unter den Namen des vorigen Loadable an einen Öffnungspunkt.
7	Drücken Sie **Umschalt + ?**, um alle verfügbaren Loadables anzuzeigen. Das Loadable XMIT wird jetzt in dieser Liste angezeigt. 

Schritt	Aktion
8	Platzieren Sie den Cursor auf XMIT, und drücken Sie die **Eingabetaste**. Im Bildschirm werden die Versionsnummer, die Größe und der Opcode des Loadable XMIT angezeigt.  Der XMIT-Opcode lautet **1e** (Hex). Hinweis: • Stellen Sie sicher, dass dieser Opcode keinen Konflikt mit anderen Opcodes verursacht, die eventuell verwendet werden. • Der im Bildschirm angezeigte Opcode kann variieren.

2.2 Installieren des XMIT-Loadable mit Concept und ProWORX

Auf einen Blick

Zweck

In diesem Abschnitt wird die Installation des Funktionsblocks für das XMIT-Loadable mit der Steuerungs-Software Concept oder ProWORX beschrieben. Vor dem Laden sollten Sie sich mit Concept bzw. ProWORX vertraut machen, je nachdem, welche Anwendung Sie verwenden.
Die Abbildungen in diesem Abschnitt dienen als Beispiele für die Bildschirme, die Sie bei der Übertragung des Loadable XMIT angezeigt bekommen.

Inhalt dieses Abschnitts

Dieser Abschnitt enthält die folgenden Themen:

Thema	Seite
Laden der NSUP- und XMIT-Loadables unter Verwendung von Concept	33
Laden der NSUP- und XMIT-Loadables unter Verwendung von ProWORX NxT	36
Laden der NSUP- und XMIT-Loadables unter Verwendung von Using ProWORX32	41

Laden der NSUP- und XMIT-Loadables unter Verwendung von Concept

Vor dem Laden

Hinweis: Ladereihenfolge

Installieren

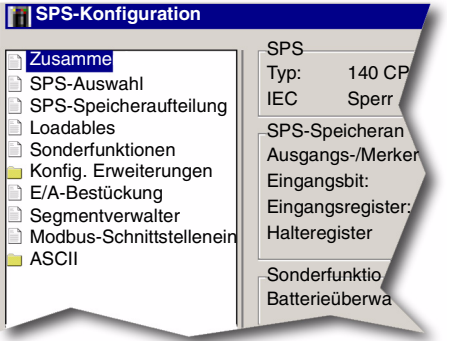
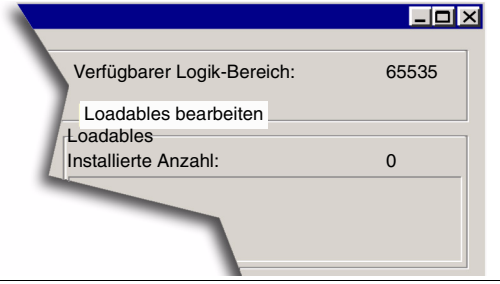
1. NSUP.EXE

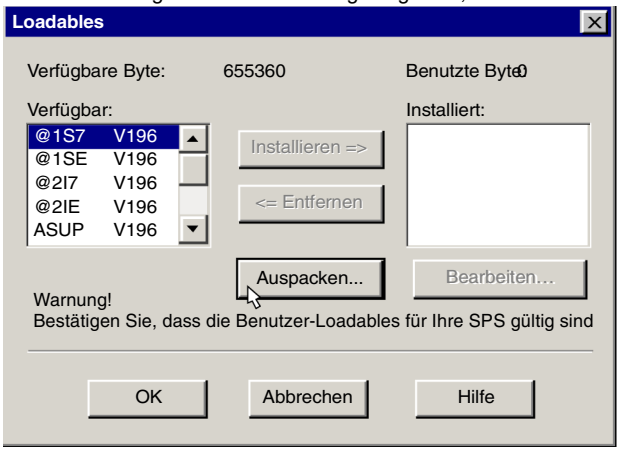
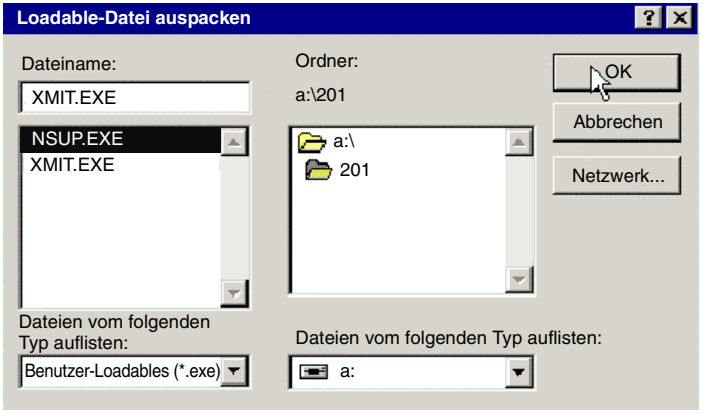
2. XMIT.EXE

Wenn die Ladevorgänge in der falschen Reihenfolge ausgeführt werden, funktioniert XMIT nicht.

Laden von NSUP.EXE

Klicken Sie im Hauptfenster der Ladder Logic in Concept auf das Dropdown-Menü für das Projekt.

Schritt	Aktion
1	Wählen Sie den Konfigurator aus. Das Dialogfeld "SPS-Konfiguration" wird angezeigt. 
2	Doppelklicken Sie im Loadables-Bereich auf "Installierte Anzahl:". 

Schritt	Aktion
3	Wenn das Dialogfeld "Loadables" angezeigt wird, klicken Sie auf "Auspacken". 
4	Klicken Sie auf OK.
5	Wenn das Dialogfeld "Loadable-Datei auspacken" angezeigt wird, wählen Sie "NSUP.EXE" aus. 
6	Klicken Sie auf OK.
7	NSUP wird jetzt im Feld mit den verfügbaren Loadables angezeigt.

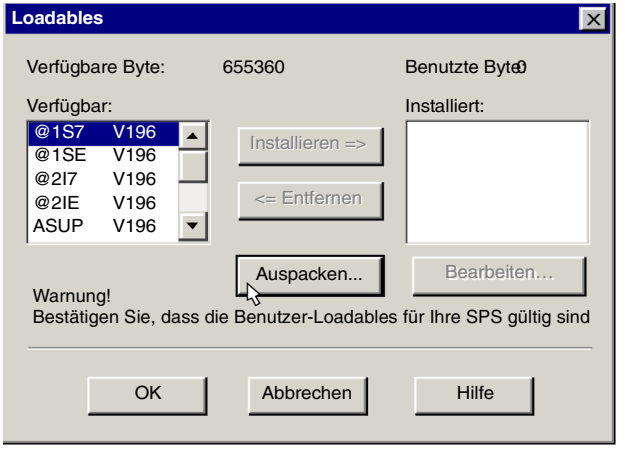
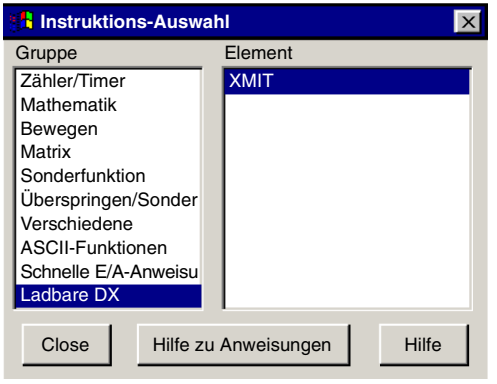
Laden von XMIT.EXE

Gehen Sie wie folgt vor, um die Datei XMIT.EXE zu auspacken:

Schritt	Aktion
1	Gehen Sie genauso vor wie beim Auspacken der Datei NSUP.EXE.

Installation der Loadables

Nachdem die Datei NSUP.EXE geladen wurde, können die Loadables in der Datenbank installiert werden, indem das entsprechende Loadable ausgewählt wird. Führen Sie dann folgende Schritte aus:

Schritt	Aktion
1	Klicken Sie auf "Install =>". 
2	Öffnen Sie den 984 LL-Editor.
3	Öffnen Sie das Dialogfeld "Instruktions-Auswahl". 
4	Das Loadable DX wird in der Gruppenliste angezeigt, und in der Elementliste wird XMIT angezeigt.
5	Klicken Sie auf Schließen.

Laden der NSUP- und XMIT-Loadables unter Verwendung von ProWORX NxT

Vor dem Laden

Hinweis: Ladereihenfolge

Installieren

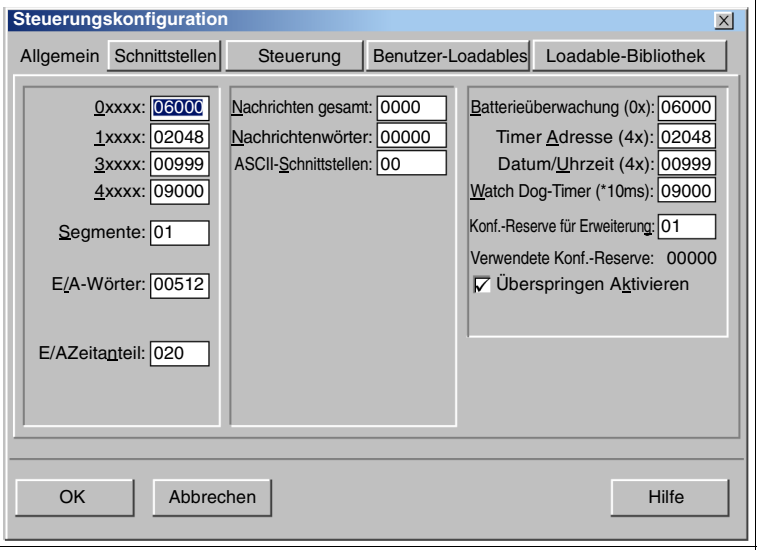
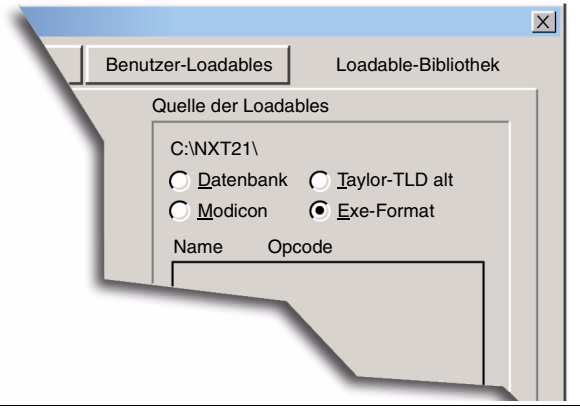
1. NSUP.EXE

2. XMIT.EXE

Wenn die Ladevorgänge in der falschen Reihenfolge ausgeführt werden, funktioniert XMIT nicht.

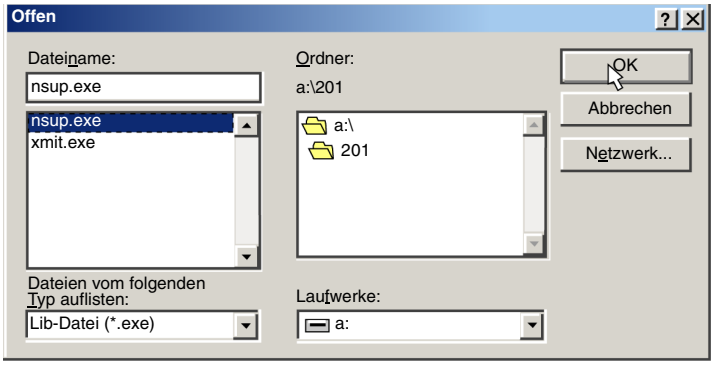
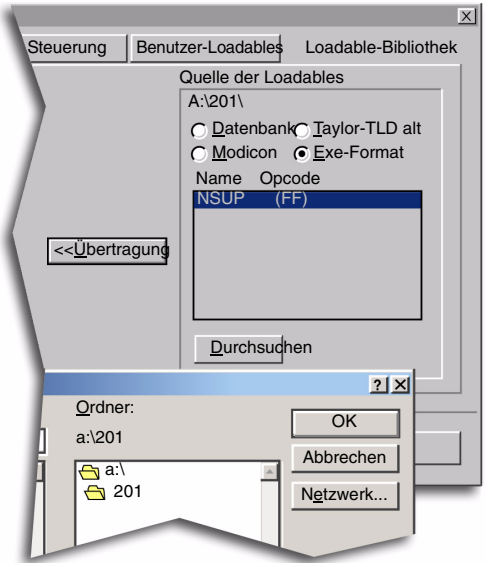
Auswahl der Datei NSUP.EXE

Klicken Sie im Hauptfenster der Ladder Logic in Proworx Nxt auf das Dropdown-Menü für die Konfiguration.

Schritt	Aktion
1	Wählen Sie die Konfigurationsoption aus. Das Dialogfeld "Steuerungskonfiguration" wird angezeigt. 
2	Klicken Sie auf die Registerkarte für die Loadable-Bibliothek.
3	Wählen Sie das Exe-Format aus. 
4	Klicken Sie auf "Durchsuchen".
5	Das Dialogfeld "Öffnen" wird angezeigt.

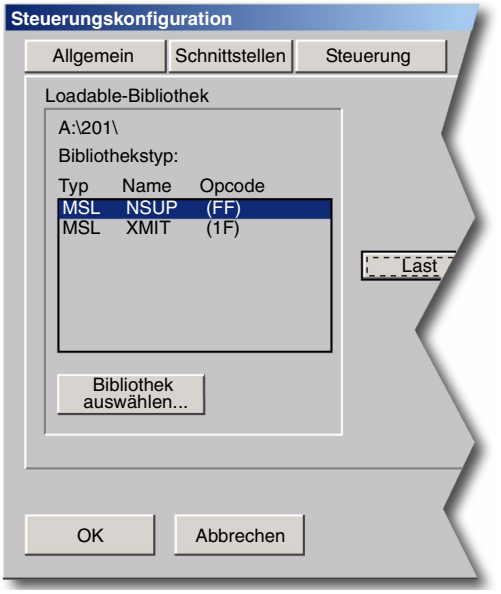
NSUP.EXE übertragen

Gehen Sie wie folgt vor, wenn das Dialogfeld "Öffnen" angezeigt wird:

Schritt	Aktion
1	Wählen Sie die Datei NSUP.EXE aus. 
2	NSUP und der zugehörige Opcode werden in der Liste "Quelle der Loadables" aufgeführt. 
3	Klicken Sie auf <<"Übertragung".
4	NSUP wird von der Liste "Quelle der Loadables" in die Liste "Loadable-Bibliothek | Bibliothekstyp" verschoben.
5	Das Dialogfeld "Öffnen" wird angezeigt. Überschreiben Sie im Dropdown-Kombinationsfeld "Dateien vom folgenden Typ auflisten:" den Ausdruck "proworx.?sl" mit dem Wort "XMIT".
6	Klicken Sie auf OK.

Laden der Datei NSUP.EXE in der Datenbank

Wählen Sie im Dialogfeld "Steuerungskonfiguration" die Registerkarte "Benutzer-Loadables" aus.

Schritt	Aktion
1	Wählen Sie in der Liste "Loadable-Bibliothek" die NSUP-Datei (MSL NSUP (FF) aus. 
2	Klicken Sie auf "Laden">>.
3	NSUP und der zugehörige OP-Code werden in die Datenbankliste der Loadables übertragen.
4	Klicken Sie auf OK. Wenn Sie auf OK klicken, gelangen Sie zurück zum Bildschirm "Ladder Logic".

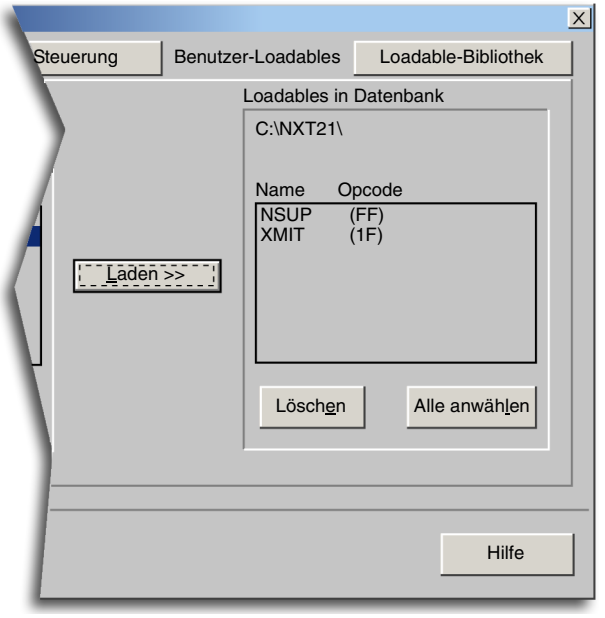
Laden von XMIT.EXE

So laden Sie XMIT.EXE:

Schritt	Aktion
1	Gehen Sie genauso vor wie beim Laden der Datei NSUP.EXE.

Fertigstellung

Gehen Sie am Ende des Ladevorgangs von NSUP.EXE und XMIT.EXE wie folgt vor:

Schritt	Aktion
1	Vergewissern Sie sich, dass sowohl NSUP.EXE (NSUP (FF)) als auch XMIT.EXE (XMIT (1F)) in der Liste der Loadables in der Datenbank angezeigt werden. 

Laden der NSUP- und XMIT-Loadables unter Verwendung von Using ProWORX32

Vor dem Laden

Hinweis: Ladereihenfolge

Installieren

1. NSUP.EXE

2. XMIT.EXE

Wenn die Ladevorgänge in der falschen Reihenfolge ausgeführt werden, funktioniert XMIT nicht.

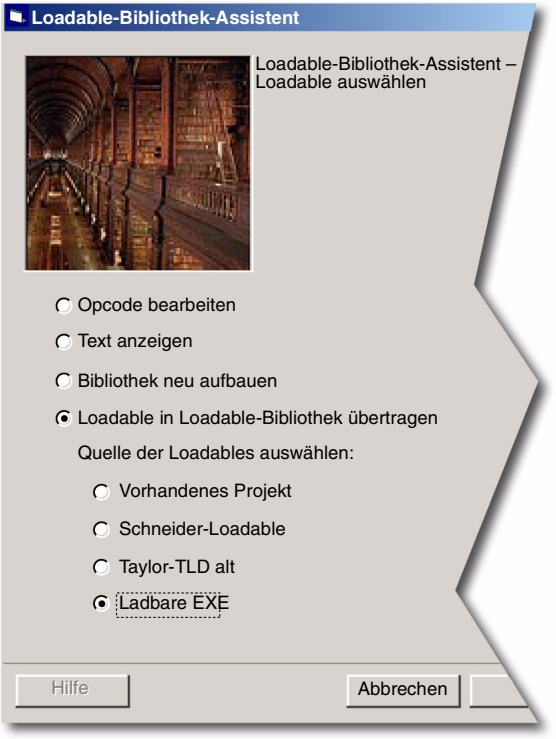
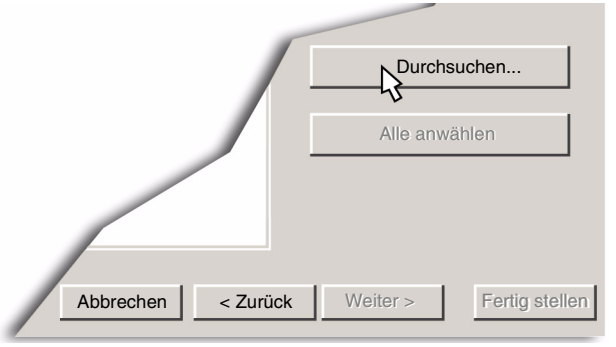
Öffnen des Assistenten

Wählen Sie im Navigationsfeld von ProWORX32 die Registerkarte "Hilfsprogramme" aus, und gehen Sie wie folgt vor:

Schritt	Aktion
1	Doppelklicken Sie auf die Option "Loadable-Bibliothek".
2	Der Loadable-Bibliothek-Assistent wird angezeigt.

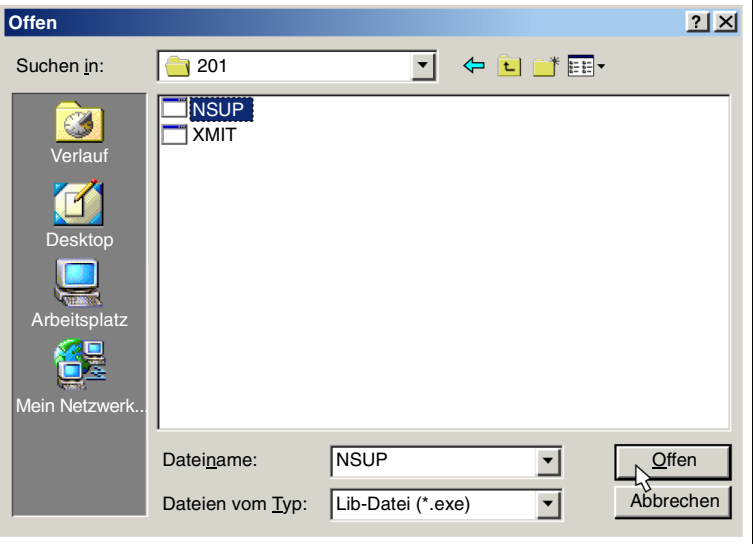
Auswahl und Übertragung von NSUP.EXE mit Hilfe des Assistenten

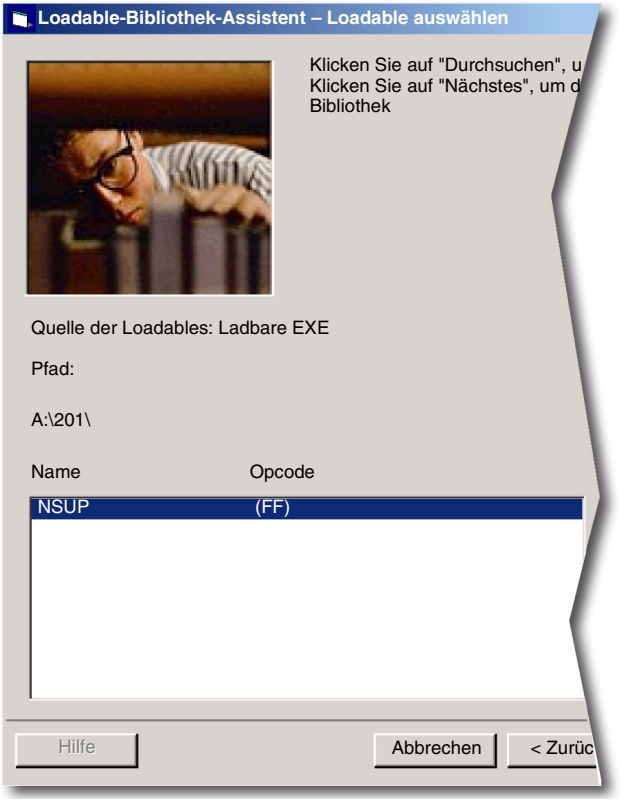
Wenn der Assistent geöffnet ist, gehen Sie wie folgt vor:

Schritt	Aktion
1	Wählen Sie die beiden Optionen "Loadable in Loadable-Bibliothek übertragen" und "EXE-Loadable" aus. 
2	Klicken Sie auf "Durchsuchen...". . 
3	NSUP wird im Pfad angezeigt: Liste
4	Wählen Sie APPLY (ANWENDEN).

Laden der Datei NSUP.EXE in der Datenbank

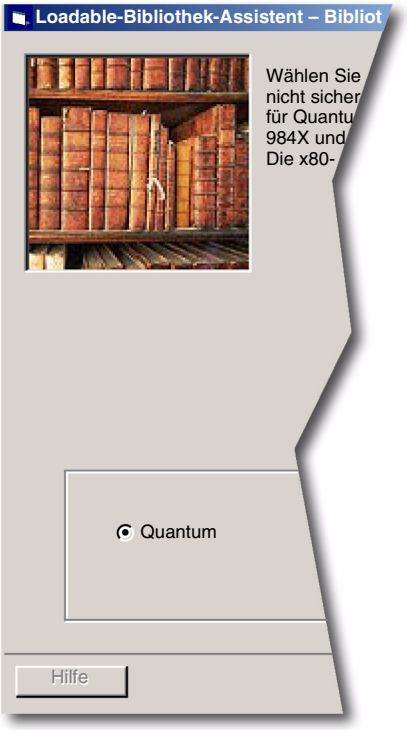
Gehen Sie wie folgt vor, wenn das Dialogfeld "Öffnen" angezeigt wird:

Schritt	Aktion
1	Wählen Sie zunächst ein NSUP-Loadable aus. 
2	Klicken Sie auf "Öffnen". Das Fenster "Loadable-Bibliothek-Assistent – Loadable auswählen" wird angezeigt.

Schritt	Aktion
3	NSUP und der zugehörige Opcode (FF) werden im Pfad angezeigt: Liste 
4	Klicken Sie auf "Nächstes".

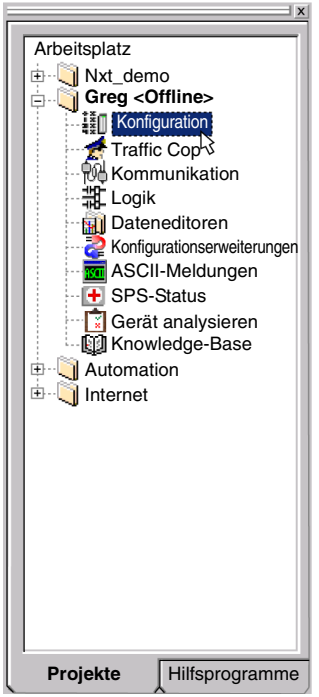
Auswahl des Bibliothekstyps

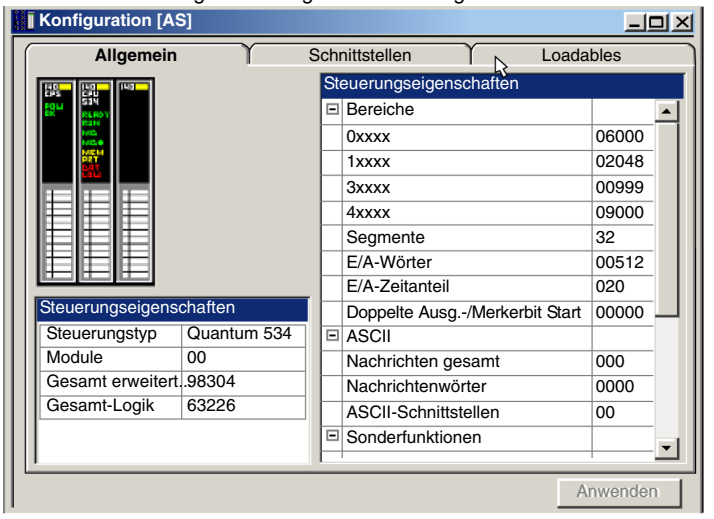
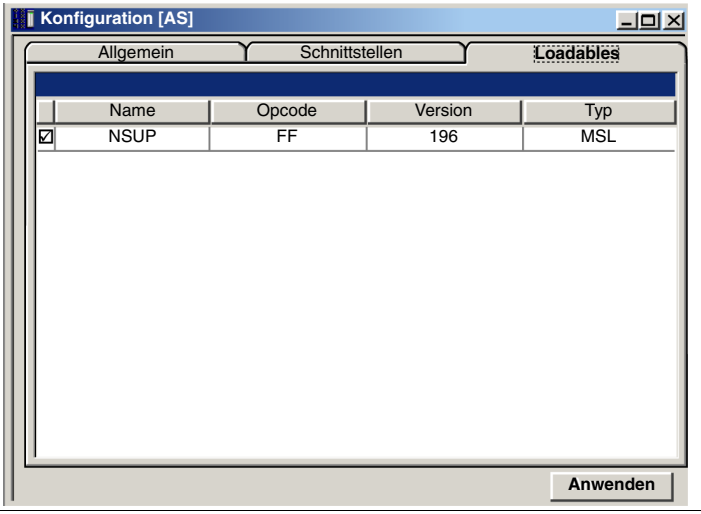
Gehen Sie wie folgt vor, wenn das Dialogfeld "Loadable-Bibliothek-Assistent – Bibliothekstyp auswählen" angezeigt wird:

Schritt	Aktion
1	Wählen Sie Quantum aus. 
2	Klicken Sie auf "Nächstes".
3	Klicken Sie auf Fertigstellen.

Konfigurieren der Steuerung

Kehren Sie ins Navigationsfeld von ProWORX32 zurück.

Schritt	Aktion
1	Wählen Sie die Registerkarte "Projekte" aus. 
2	Klicken Sie auf die Option "Konfiguration".

Schritt	Aktion
3	Wählen Sie im Dialogfeld "Konfiguration" die Registerkarte "Loadables" aus. 
4	Aktivieren Sie das Kontrollkästchen "NSUP". 
5	Klicken Sie auf "Anwenden".

Laden von XMIT.EXE

So laden Sie XMIT.EXE:

Schritt	Aktion
1	Gehen Sie genauso vor wie beim Laden der Datei NSUP.EXE.

Verwendung von Zoomfenstern

3

Auf einen Blick

Zweck

In diesem Fenster wird die Verwendung der Zoomfenster in Modsoft, ProWORX NxT und ProWORX32 beschrieben. In Concept sind keine Zoomfenster verfügbar. Verwenden Sie die Zoomfenster zum Konfigurieren der Parameter des ladbaren oder integrierten XMIT-Blocks. Darüber hinaus können Sie die Zoomfenster verwenden, um die Register für die Kommunikation, den Schnittstellenstatus oder Konvertierungsblöcke zu konfigurieren.

Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Thema	Seite
Zoomfenster für den Kommunikationsblock in Modsoft	50
Zoomfenster für den Schnittstellenstatusblock in Modsoft	54
Zoomblöcke für Konvertierungsblocks in Modsoft	56
Zoomfenster, in denen Concept verwendet wird	58
Zoomfenster, in denen ProWORX NxT verwendet wird	62
Zoomfenster, in denen ProWORX32 verwendet wird	65

Zoomfenster für den Kommunikationsblock in Modsoft

Überblick

In diesem Abschnitt werden die Zoomfenster für Modsoft DX beschrieben, die dem Kommunikationsblock des XMIT-Funktionsblocks zugeordnet sind. Im Kommunikationsmodus sind acht (8) Zoomfenster verfügbar. Sie dienen der Konfiguration der Parameter in Register 4x bis 4x + 15, der Anzeige der ASCII-Eingangsinformationen, der Informationen zum Offset für den Zeiger und der Anzeige von Fehlerstatusinformationen.

Zoomfenster für die Kommunikation (acht)

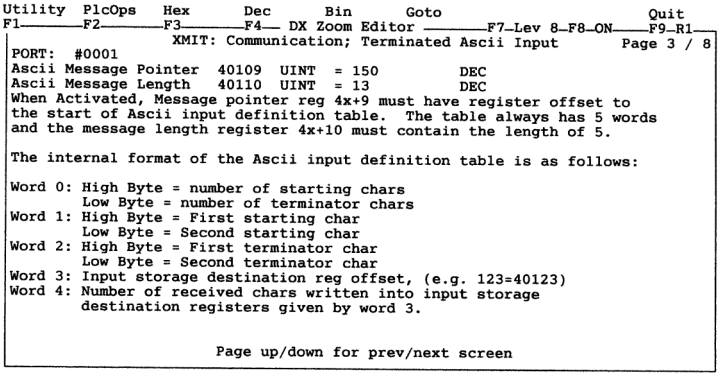
DX-Zoomfenster für die Kommunikation, Seite 1/8, wird für die Konfiguration der Register 4x bis 4x + 8 verwendet.

```
Utility          Hex      Dec      Bin      Goto                        Quit
F1———MIKE———F3———F4— DX Zoom Editor ———F7—Lev 8—F8—OFF—F9
                          XMIT: Communication                  Page 1 / 8

4X   XMIT Revision Number                          40100  UINT  = 201          DEC
4X+1 Fault Status                                  40101  UINT  = 0            DEC
4X+2 Available To User:                            40102  UINT  = 0000         HEX
4X+3 Data Rate(50,75,110,134,150,300,600,1200,
     1800,2000,2400,3600,4800,7200,9600,19200)     40103  UINT  = 9600         DEC
4X+4 Data Bits (7,8):                              40104  UINT  = 8            DEC
4X+5 Parity (0=none; 1=odd; 2=even):               40105  UINT  = 2            DEC
4X+6 Stop Bits (1,2):                              40106  UINT  = 1            DEC
4X+7 Available To User:                            40107  UINT  = 0000         HEX
4X+8 Command Word:                                 40108  UINT  = 0100000000000000

B2 RTS/CTS Modm Ctrl:    ENABLED       | B3 RS485 MODE:          DISABLED
B5 Term'd Asc Input:     NO            | B6 Simple Asc Input:    NO
B7 ASCII Strng Msging:   DISABLED      | B8 Modbus Msging:       DISABLED
B9 Asc Recv FIFO:        DISABLED      | B10 Backspace:          DISABLED
B11 RTS/CTS Flow Ctrl:   DISABLED      | B12 Xon/Xoff Flow Ctrl:DISABLED
B13 Puls Dial Modm ATDP: NO            | B14 Hang-up Modm AT :   NO
B15 Tone Dial Modm ATDT: NO            | B16 Init Modem AT:      NO

                      Page up/down for prev/next screen
```

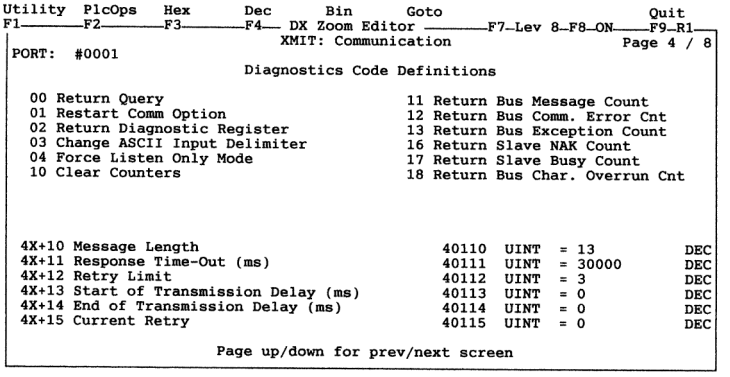
Das DX-Zoomfenster für die Kommunikation, Seite 2/8, wird für die Konfiguration des Registers 4x + 9 verwendet (Nachrichtenzeiger).

```
Utility  PlcOps   Hex      Dec      Bin      Goto                        Quit
F1———F2———F3———F4— DX Zoom Editor ———F7—Lev 8—F8—ON—F9—R1—
                          XMIT: Communication                  Page 2 / 8
PORT:  #0001
 4X+9 Message Pointer                          40109  UINT  = 150          DEC

 Function 01-06,15,16    Modbus function 08        Modbus functions 20,21
  4Y   Function code       4Y   Function code        4Y   Function code
  4Y+1 Quantity            4Y+1 Subfunction Code     4Y+1 Quantity
  4Y+2 Slave PLC address   4Y+2 Slave PLC address    4Y+2 Slave PLC address
  4Y+3 Slave data area     4Y+3 Diag. funct. data    4Y+3 Slave data area
  4Y+4 Master data area    4Y+4 Master data area     4Y+4 Master data area
                                                     4Y+5 File number

    Modbus Function Code Definitions
  01 Read Coil Status                02 Read Input Status
  03 Read Holding Registers          04 Read Input Registers
  05 Force Single Coil               06 Preset Single Register
  08 Diagnostics
  15 Force Multiple Coils            16 Preset Multiple Registers
  20 Read general Reference (6X)     21 Write General Reference (6X)

                      Page up/down for prev/next screen
```

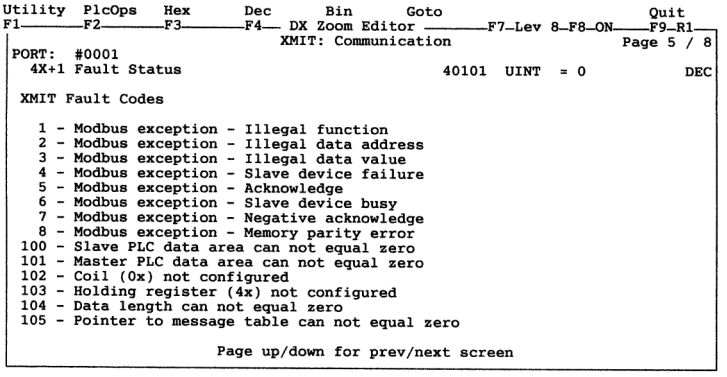
DX-Zoomfenster für die Kommunikation, Seite 3/8, wird verwendet, um die ASCII-Eingangsinformationen anzuzeigen, wenn das Offset für Register 4x + 9 auf die Definitionstabelle für den ASCII-Eingang verweist.


Utility PlcOps Hex Dec Bin Goto Quit
F1 F2 F3 F4 DX Zoom Editor F7 Lev 8 F8 ON F9 R1
XMIT: Communication; Terminated Ascii Input Page 3 / 8
PORT: #0001
Ascii Message Pointer 40109 UINT = 150 DEC
Ascii Message Length 40110 UINT = 13 DEC
When Activated, Message pointer reg 4x+9 must have register offset to
the start of Ascii input definition table. The table always has 5 words
and the message length register 4x+10 must contain the length of 5.
The internal format of the Ascii input definition table is as follows:
Word 0: High Byte = number of starting chars
Low Byte = number of terminator chars
Word 1: High Byte = First starting char
Low Byte = Second starting char
Word 2: High Byte = First terminator char
Low Byte = Second terminator char
Word 3: Input storage destination reg offset, (e.g. 123=40123)
Word 4: Number of received chars written into input storage
destination registers given by word 3.
Page up/down for prev/next screen

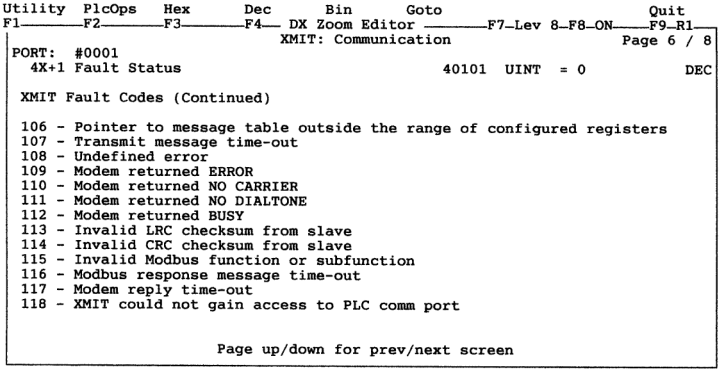
Das DX-Zoomfenster für die Kommunikation, Seite 4/8, wird verwendet, um (a) die Definitionen der Diagnosecodes anzuzeigen und (b) die Register 4x + 10 bis 4x + 15 zu konfigurieren.


Utility PlcOps Hex Dec Bin Goto Quit
F1 F2 F3 F4 DX Zoom Editor F7 Lev 8 F8 ON F9 R1
XMIT: Communication Page 4 / 8
PORT: #0001
Diagnostics Code Definitions
00 Return Query
01 Restart Comm Option
02 Return Diagnostic Register
03 Change ASCII Input Delimiter
04 Force Listen Only Mode
10 Clear Counters
11 Return Bus Message Count
12 Return Bus Comm. Error Cnt
13 Return Bus Exception Count
16 Return Slave NAK Count
17 Return Slave Busy Count
18 Return Bus Char. Overrun Cnt
4X+10 Message Length 40110 UINT = 13 DEC
4X+11 Response Time-Out (ms) 40111 UINT = 30000 DEC
4X+12 Retry Limit 40112 UINT = 3 DEC
4X+13 Start of Transmission Delay (ms) 40113 UINT = 0 DEC
4X+14 End of Transmission Delay (ms) 40114 UINT = 0 DEC
4X+15 Current Retry 40115 UINT = 0 DEC
Page up/down for prev/next screen

Das DX-Zoomfenster für die Kommunikation, Seite 5/8, zeigt den Fehlerstatus der XMIT-Kommunikation (Fehlercodes 1–8 und 100–105) an.


Utility PlcOps Hex Dec Bin Goto Quit
F1 F2 F3 F4 DX Zoom Editor F7 Lev 8 F8 ON F9 R1
XMIT: Communication Page 5 / 8
PORT: #0001
4X+1 Fault Status 40101 UINT = 0 DEC
XMIT Fault Codes
1 - Modbus exception - Illegal function
2 - Modbus exception - Illegal data address
3 - Modbus exception - Illegal data value
4 - Modbus exception - Slave device failure
5 - Modbus exception - Acknowledge
6 - Modbus exception - Slave device busy
7 - Modbus exception - Negative acknowledge
8 - Modbus exception - Memory parity error
100 - Slave PLC data area can not equal zero
101 - Master PLC data area can not equal zero
102 - Coil (0x) not configured
103 - Holding register (4x) not configured
104 - Data length can not equal zero
105 - Pointer to message table can not equal zero
Page up/down for prev/next screen

Das DX-Zoomfenster für die Kommunikation, Seite 6/8, zeigt die Fehlermeldungen für die XMIT-Kommunikation (Fehlercodes 106–118) an.


Utility PlcOps Hex Dec Bin Goto Quit
F1 F2 F3 F4 DX Zoom Editor F7 Lev 8 F8 ON F9 R1
XMIT: Communication Page 6 / 8
PORT: #0001
4X+1 Fault Status 40101 UINT = 0 DEC
XMIT Fault Codes (Continued)
106 - Pointer to message table outside the range of configured registers
107 - Transmit message time-out
108 - Undefined error
109 - Modem returned ERROR
110 - Modem returned NO CARRIER
111 - Modem returned NO DIALTONE
112 - Modem returned BUSY
113 - Invalid LRC checksum from slave
114 - Invalid CRC checksum from slave
115 - Invalid Modbus function or subfunction
116 - Modbus response message time-out
117 - Modem reply time-out
118 - XMIT could not gain access to PLC comm port
Page up/down for prev/next screen

Das DX-Zoomfenster für die Kommunikation, Seite 7/8, zeigt den Fehlerstatus der XMIT-Kommunikation (Fehlercodes 119–131) an.

```
Utility  PlcOps   Hex      Dec      Bin      Goto                        Quit
F1———————F2———————F3———————F4—— DX Zoom Editor ———————F7—Lev 8—F8—ON———F9—R1—
                              XMIT: Communication                Page 7 / 8
PORT:  #0001
  4X+1 Fault Status                              40101  UINT  = 0          DEC

 XMIT Fault Codes (Continued)

 119 - XMIT could not enable PLC port receiver
 120 - XMIT could not set PLC UART
 121 - User issued an abort command
 122 - Top node of XMIT is not equal to one
 123 - Bottom node of XMIT is not equal to sixteen
 124 - Undefined internal state
 125 - Broadcast mode not allowed with this Modbus function
 126 - DCE did not assert CTS
 127 - Illegal configuration(Data bits, Data rate, Parity or Stop bits)
 128 - Unexpected response received from Modbus slave
 129 - Invalid command word combination
 130   Command word changed while active
 131 - Invalid character count


                        Page up/down for prev/next screen
```

Das DX-Zoomfenster für die Kommunikation, Seite 8/8, zeigt den Fehlerstatus der XMIT-Kommunikation (Fehlercodes 132–143) an.

```
Utility  PlcOps   Hex      Dec      Bin      Goto                        Quit
F1———————F2———————F3———————F4—— DX Zoom Editor ———————F7—Lev 8—F8—ON———F9—R1—
                              XMIT: Communication                Page 8 / 8
PORT:  #0001
  4X+1 Fault Status                              40101  UINT  = 0          DEC

 XMIT Fault Codes (Continued)

 132 - Invalid register block
 133 - Ascii input FIF0 overflow error
 134 - Invalid number of start chars or termination chars
 135 - Invalid destination register block
 136 - Invalid source register block
 137 - No Ascii number present
 138 - Illegal configuration(Data bits, Data rate, Parity or Stop bits)
 139 - Numerical overflow detected
 140 - String mismatch error
 141 - String not found error
 142 - Invalid error check detected
 143 - Invalid conversion Opcode



                             END XMIT COMMUNICATION
```

Zoomfenster für den Schnittstellenstatusblock in Modsoft

Zoomfenster für den Schnittstellenstatus (Drei)

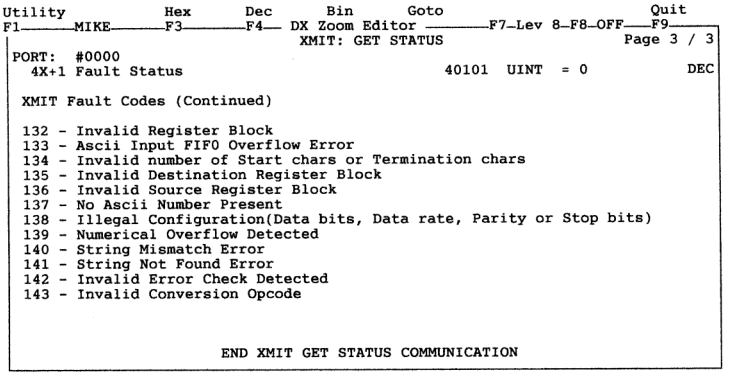
Das DX-Zoomfenster für den Schnittstellenstatus, Seite 1/3, wird zur Konfiguration des Statusabrufs verwendet.

```
Utility          Hex      Dec      Bin      Goto                          Quit
F1———MIKE———F3———F4— DX Zoom Editor ———F7—Lev 8—F8—OFF—F9———
                          XMIT: GET STATUS                    PAGE 1 OF 3
PORT:  #0000
XMIT Revision Number         :40100  UINT  = 201          DEC

Fault Status                 :40101  UINT  = 0            DEC

Slave Logged-In (0=No;1=Yes) :40102  01:08 = 0            DEC
Slave Active(0=No;1=Yes)     :40102  09:16 = 0            DEC
Slave Transaction Counter    :40103  UINT  = 9600         DEC

Port State                   :40104  01:16 = 8            DEC

  Input FIFO Status Bits     :40105  UINT  = 0000000000000010
  Port Owned By(0=PLC;1=XMIT)             : 40105  04:04 = 0          DEC
  Ascii Output Blocked By Recvr(0=No;1=Yes) : 40105  08:08 = 0          DEC
  Ascii Inpt Has Blockd Send Dev(0=No;1=Yes): 40105  16:16 = 0          DEC
  Ascii Input FIFO Rcvd New Char(0=No;1=Yes): 40105  09:09 = 0          DEC
  Ascii Input FIFO Empty(0=No;1=Yes)      : 40105  10:10 = 0          DEC
  Ascii Input FIFO Enabled()=No;1=Yes)    : 40105  12:12 = 0          DEC
  Input FIFO Overflow Error(0=No;1=Yes)   : 40105  11:11 = 0          DEC
  Input FIFO Length                       : 40106  UINT  = 1          DEC
```

Das DX-Zoomfenster für den Schnittstellenstatus, Seite 2/3, zeigt Fehlermeldungen des XMIT-Schnittstellenstatus (Fehlercodes 119–131) an.

```
Utility          Hex      Dec      Bin      Goto                          Quit
F1———MIKE———F3———F4— DX Zoom Editor ———F7—Lev 8—F8—OFF—F9———
                          XMIT: GET STATUS                    Page 2 / 3
PORT:  #0000
  4X+1 Fault Status                        40101  UINT  = 0          DEC

 XMIT Fault Codes (Continued)

 119 - XMIT Could Not Enable PLC Port Receiver
 120 - XMIT Could Not Set PLC UART
 121 - User Issued an Abort Command
 122 - Top node of XMIT is not equal to one
 123 - Bottom Node of XMIT is not equal to sixteen
 124 - Undefined Internal state
 125 - Broadcast Mode not Allowed with this Modbus Function
 126 - DCE did not Assert CTS
 127 - Illegal configuration(Data bits, Data rate, Parity or Stop bits)
 128 - Unexpected Response Received from Modbus Slave
 129 - Invalid Command Word Combination
 130 - Command Word Changed While Active
 131 - Invalid Character Count


                     Page up/down for prev/next screen
```

Das DX-Zoomfenster für den Schnittstellenstatus, Seite 3/3, zeigt Fehlermeldungen des XMIT-Schnittstellenstatus (Fehlercodes 132–143) an.


Utility Hex Dec Bin Goto Quit
F1 MIKE F3 F4 DX Zoom Editor F7-Lev 8-F8-OFF F9
XMIT: GET STATUS Page 3 / 3
PORT: #0000
4X+1 Fault Status 40101 UINT = 0 DEC
XMIT Fault Codes (Continued)
132 - Invalid Register Block
133 - Ascii Input FIFO Overflow Error
134 - Invalid number of Start chars or Termination chars
135 - Invalid Destination Register Block
136 - Invalid Source Register Block
137 - No Ascii Number Present
138 - Illegal Configuration(Data bits, Data rate, Parity or Stop bits)
139 - Numerical Overflow Detected
140 - String Mismatch Error
141 - String Not Found Error
142 - Invalid Error Check Detected
143 - Invalid Conversion Opcode
END XMIT GET STATUS COMMUNICATION

Zoomblöcke für Konvertierungsblocks in Modsoft

Überblick

In diesem Abschnitt werden die Zoomfenster für Modsoft DX beschrieben, die dem Konvertierungsblock des XMIT-Funktionsblocks zugeordnet sind. Im Konvertierungsmodus sind drei (3) Zoomfenster zur Konfiguration und zur Anzeige von Fehlerstatusinformationen verfügbar.

Zoomfenster für die Konvertierung (Drei)

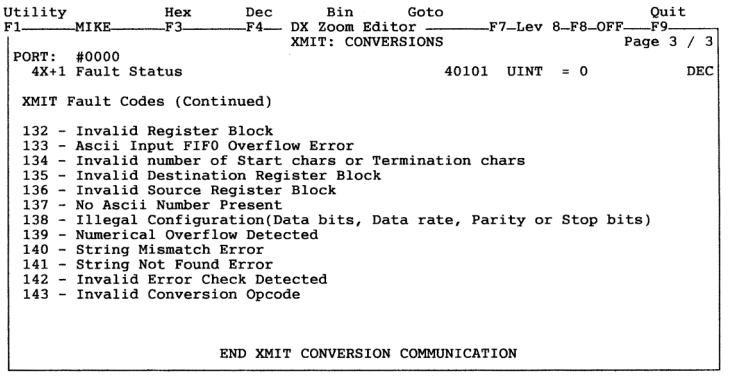
Das DX-Zoomfenster für die Konvertierung, Seite 1/3, wird für die Konfiguration von Konvertierungen verwendet.

```
Utility          Hex      Dec      Bin      Goto                        Quit
F1———MIKE———F3———F4— DX Zoom Editor ———F7—Lev 8—F8—OFF—F9———
|                          XMIT: CONVERSIONS                  PAGE 1 OF 3

XMIT Revision Number:          40100  UINT  = 201          DEC
Fault Status:                  40101  UINT  = 0            DEC
Register Avail to User:        40102  UINT  = 0000         HEX
Data Conversion OpCode:                      COPY SRC BLK->DEST

Data Conversion Control Bits: 40103  UINT  = 0010010110000000
B2: CRC Seed:        0xFFFF          | B3: Err Chk Type:    LRC-8
B4: Error Check:     APPEND          |
B7: Conv. Case:      UPPR->LOWR      | B8: Case Sens.       NO
B9: Format Leading:  ZEROS           | B10: Output Format:  VARIABLE
B11: Conv. Type      SIGNED          | B12: Conv. Word:     16 BIT
B13: Auto Adv Src:   NO              | B14: Auto Adv Dest:  NO
B15: Begin Read Src: HI BYTE         | B16: Begin Save Dest:HI BYTE

SRC Register Offset :40105  UINT  = 2          DEC     Begin Read At:HI BYTE
                     (e.g. 100->40100)(Must be non-zero)
DEST Register Offset:40106  UINT  = 1          DEC     Begin Save At: HI BYTE
                     (e.g. 200->40200)(Must be non-zero)
Data Conversion Character Count:   40107  UINT  = 0          DEC
```

Im DX-Zoomfenster für die Konvertierung, Seite 2/3, werden Fehlermeldungen bei der XMIT-Konvertierung angezeigt (Fehlercodes 119–131).

```
Utility          Hex      Dec      Bin      Goto                        Quit
F1———MIKE———F3———F4— DX Zoom Editor ———F7—Lev 8—F8—OFF—F9———
|                          XMIT: CONVERSIONS                  Page 2 / 3
PORT:  #0000
  4X+1 Fault Status                          40101  UINT  = 0            DEC

 XMIT Fault Codes (Continued)

 119 - XMIT Could Not Enable PLC Port Receiver
 120 - XMIT Could Not Set PLC UART
 121 - User Issued an Abort Command
 122 - Top node of XMIT is not equal to one
 123 - Bottom Node of XMIT is not equal to sixteen
 124 - Undefined Internal state
 125 - Broadcast Mode not Allowed with this Modbus Function
 126 - DCE did not Assert CTS
 127 - Illegal configuration(Data bits, Data rate, Parity or Stop bits)
 128 - Unexpected Response Received from Modbus Slave
 129 - Invalid Command Word Combination
 130 - Command Word Changed While Active
 131 - Invalid Character Count


                      Page up/down for prev/next screen
```

Im DX-Zoomfenster für die Konvertierung, Seite 3/3, werden die Fehlermeldungen bei der XMIT-Konvertierung angezeigt (Fehlercodes 132–143).


Utility Hex Dec Bin Goto Quit
F1 MIKE F3 F4 DX Zoom Editor F7 Lev 8 F8 OFF F9
XMIT: CONVERSIONS Page 3 / 3
PORT: #0000
4X+1 Fault Status 40101 UINT = 0 DEC
XMIT Fault Codes (Continued)
132 - Invalid Register Block
133 - Ascii Input FIFO Overflow Error
134 - Invalid number of Start chars or Termination chars
135 - Invalid Destination Register Block
136 - Invalid Source Register Block
137 - No Ascii Number Present
138 - Illegal Configuration(Data bits, Data rate, Parity or Stop bits)
139 - Numerical Overflow Detected
140 - String Mismatch Error
141 - String Not Found Error
142 - Invalid Error Check Detected
143 - Invalid Conversion Opcode
END XMIT CONVERSION COMMUNICATION

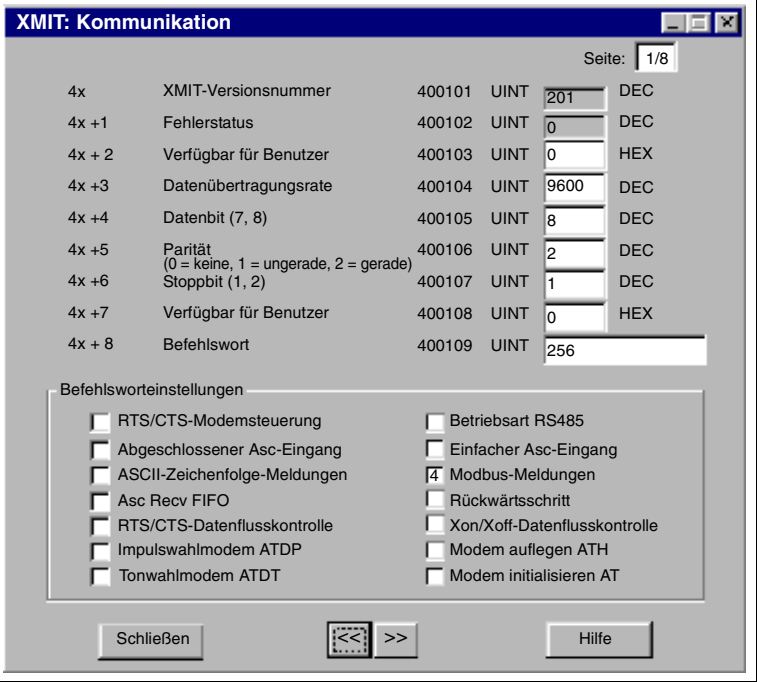
Zoomfenster, in denen Concept verwendet wird

Überblick

In diesem Abschnitt wird die Verwendung der Zoomfenster in der Steuerungs-Software Concept zur Konfiguration von Parametern und Registern beschrieben.

Zugriff auf die Zoomfenster in Concept

Ausgehend vom Ladder Logic-Editor:

Schritt	Aktion
1	Platzieren Sie einen XMIT-Block im Logikbereich.
2	In den oberen, mittleren und unteren Eintrag müssen gültige Einträge eingegeben werden. Geben Sie beispielsweise #0001 in den oberen, 400001 in den mittleren und #00016 in den unteren Eintrag ein.
3	Platzieren Sie den Cursor auf dem XMIT-Block.
4	Drücken Sie "STRG + D". Seite 1/8 in XMIT: Das Dialogfeld "Kommunikation" wird angezeigt 
5	Geben Sie die erforderlichen Parameter ein.
6	Klicken Sie auf die Schaltfläche zum Erweitern (>>), um die Seiten 2 bis 8 aufzurufen, oder klicken Sie auf "Schließen".

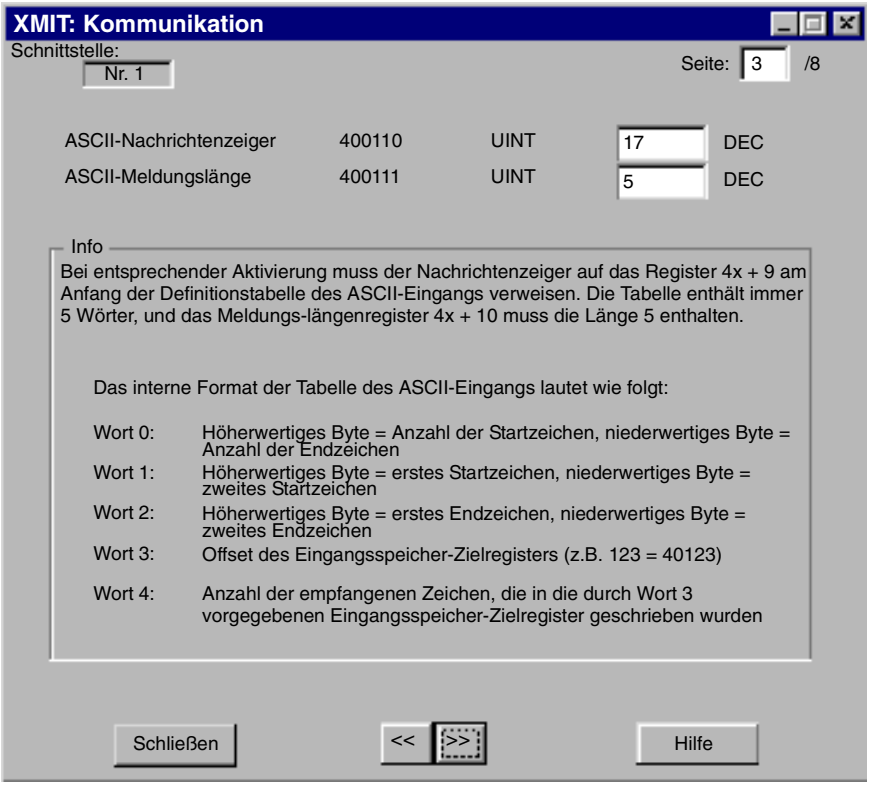
Umschalten zwischen den Zoomfenstern

Es sind acht (8) Zoomfenster verfügbar. Die Seiten 1 bis 4 sind Konfigurationsbildschirme. Die Seiten 5 bis 8 sind Fehlerstatusbildschirme.

Schritt	Aktion
1	Wählen Sie die Schaltfläche "Nächste" oder "Zurück" aus (<< >>), um zwischen den acht Fenstern umzuschalten.

Zoomfenster für die XMIT-Kommunikation 2/8

XMIT: Zoomfenster für die Kommunikation 3/8


XMIT: Kommunikation
Schnittstelle:
Nr. 1
Seite: 3 /8
ASCII-Nachrichtenzeiger 400110 UINT 17 DEC
ASCII-Meldungslänge 400111 UINT 5 DEC
Info
Bei entsprechender Aktivierung muss der Nachrichtenzeiger auf das Register 4x + 9 am Anfang der Definitionstabelle des ASCII-Eingangs verweisen. Die Tabelle enthält immer 5 Wörter, und das Meldungs-längenregister 4x + 10 muss die Länge 5 enthalten.
Das interne Format der Tabelle des ASCII-Eingangs lautet wie folgt:
Wort 0: Höherwertiges Byte = Anzahl der Startzeichen, niederwertiges Byte = Anzahl der Endzeichen
Wort 1: Höherwertiges Byte = erstes Startzeichen, niederwertiges Byte = zweites Startzeichen
Wort 2: Höherwertiges Byte = erstes Endzeichen, niederwertiges Byte = zweites Endzeichen
Wort 3: Offset des Eingangsspeicher-Zielregisters (z.B. 123 = 40123)
Wort 4: Anzahl der empfangenen Zeichen, die in die durch Wort 3 vorgegebenen Eingangsspeicher-Zielregister geschrieben wurden
Schließen
<<
>>
Hilfe

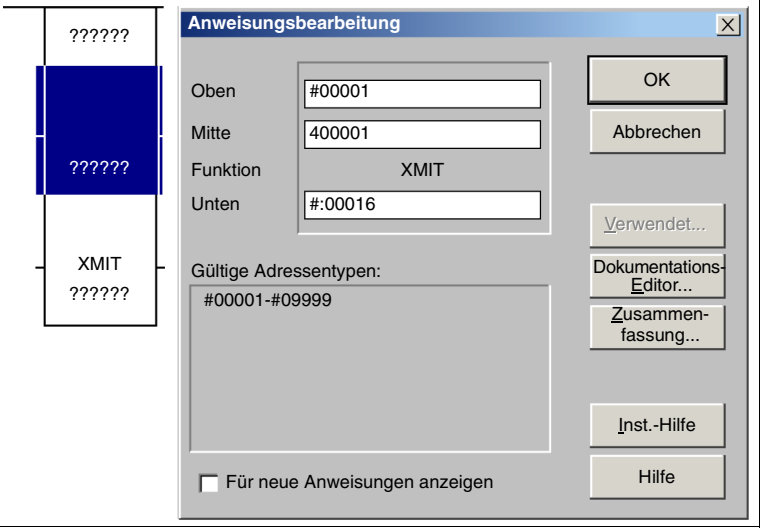
XMIT: Zoomfenster für die Kommunikation 4/8

XMIT: Kommunikation
Schnittstelle:
Nr. 1
Seite: 4 /8
4x + 10 Nachrichtenlänge 400111 UINT 5 DEC
4x + 11 Antwort-Zeitüberschreitung (ms) 400112 UINT 1000 DEC
4x + 12 Wiederholungsgrenze 400113 UINT 5 DEC
4x + 13 Beginn der Übertragungsverzögerung (ms) 400114 UINT 100 DEC
4x + 14 Ende der Übertragungsverzögerung (ms) 400115 UINT 100 DEC
4x + 15 Aktuelle Wiederholung 400116 UINT 0 DEC
Diagnosecode-Funktionen
00 Anfrage zurücksenden
01 Kommunikationsoption neu starten
02 Diagnoseregister zurücksenden
03 ASCII-Eingangsabgrenzung ändern
04 Listen-Only-Modus erzwingen
10-Register löschen
11 Bus-Nachrichten-Count zurücksenden
12 Bus-Nachrichten-Komm. zurücksenden Fehlerzählung
13 Bus-Ausnahme-Count zurücksenden
16 Slave-NAK-Zähler zurücksenden
17 Slave-Besetzt-Zähler zurücksenden
18 Bus-Nachrichten-Zeichen zurücksenden Überlauf-Count
Schließen
<<
>>
Hilfe

Zoomfenster, in denen ProWORX NxT verwendet wird

Zugriff auf die Zoomfenster in ProWORX NxT

Ausgehend vom Ladder Logic-Editor:

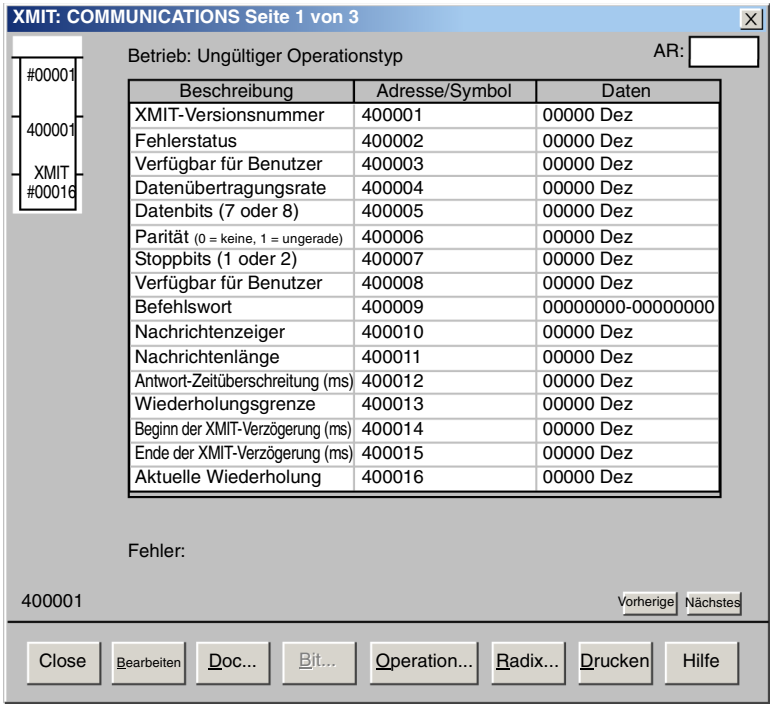
Schritt	Aktion
1	Platzieren Sie einen XMIT-Block im Logikbereich.
2	Das Dialogfeld "Anweisung bearbeiten" wird angezeigt.
3	In den oberen, mittleren und unteren Eintrag müssen gültige Einträge eingegeben werden. Geben Sie beispielsweise #0001 in den oberen, 400001 in den mittleren und #00016 in den unteren Eintrag ein.
4	Klicken Sie auf OK.
5	Platzieren Sie den Cursor auf dem XMIT-Block. 
6	Drücken Sie "Strg + R". Daraufhin werden die Zoomfenster angezeigt.

Umschalten zwischen den Zoomfenstern

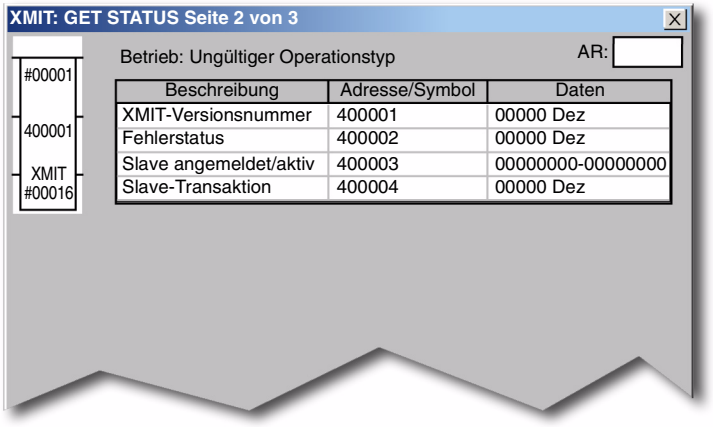
Es sind drei (3) Zoomfenster verfügbar.

Schritt	Aktion
1	Wählen Sie die Schaltfläche "Vorherige" oder "Nächstes", um zwischen den Fenstern umzuschalten.

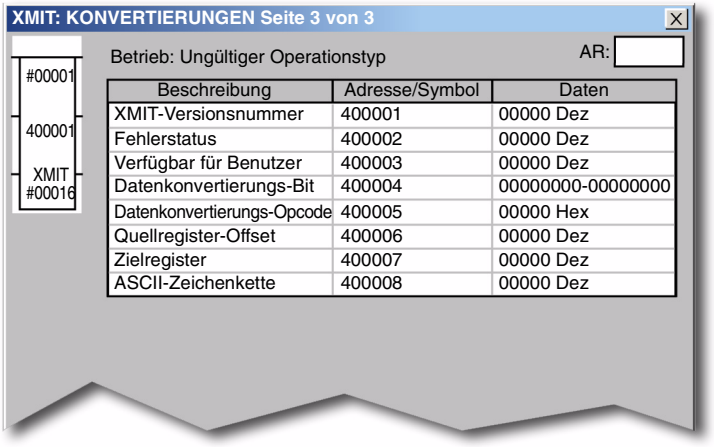
Zoomfenster für die Kommunikation


XMIT: COMMUNICATIONS Seite 1 von 3
#00001
400001
XMIT
#00016
Betrieb: Ungültiger Operationstyp
AR:
Beschreibung | Adresse/Symbol | Daten
XMIT-Versionsnummer | 400001 | 00000 Dez
Fehlerstatus | 400002 | 00000 Dez
Verfügbar für Benutzer | 400003 | 00000 Dez
Datenübertragungsrate | 400004 | 00000 Dez
Datenbits (7 oder 8) | 400005 | 00000 Dez
Parität (0 = keine, 1 = ungerade) | 400006 | 00000 Dez
Stoppbits (1 oder 2) | 400007 | 00000 Dez
Verfügbar für Benutzer | 400008 | 00000 Dez
Befehlswort | 400009 | 00000000-00000000
Nachrichtenzeiger | 400010 | 00000 Dez
Nachrichtenlänge | 400011 | 00000 Dez
Antwort-Zeitüberschreitung (ms) | 400012 | 00000 Dez
Wiederholungsgrenze | 400013 | 00000 Dez
Beginn der XMIT-Verzögerung (ms) | 400014 | 00000 Dez
Ende der XMIT-Verzögerung (ms) | 400015 | 00000 Dez
Aktuelle Wiederholung | 400016 | 00000 Dez
Fehler:
400001
Vorherige
Nächstes
Close
Bearbeiten
Doc...
Bit...
Operation...
Radix...
Drucken
Hilfe

Zoomfenster für den Statusabruf



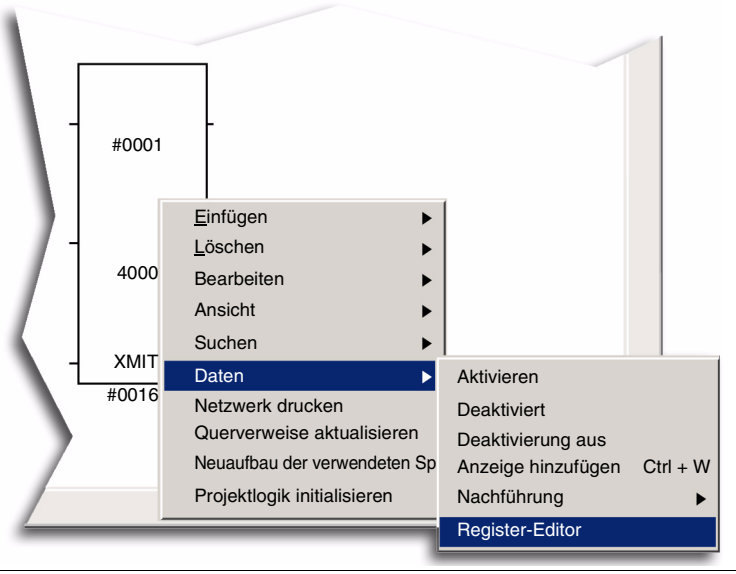
Zoomfenster für die Konvertierung


XMIT: KONVERTIERUNGEN Seite 3 von 3
#00001
400001
XMIT
#00016
Betrieb: Ungültiger Operationstyp
AR:
Beschreibung | Adresse/Symbol | Daten
XMIT-Versionsnummer | 400001 | 00000 Dez
Fehlerstatus | 400002 | 00000 Dez
Verfügbar für Benutzer | 400003 | 00000 Dez
Datenkonvertierungs-Bit | 400004 | 00000000-00000000
Datenkonvertierungs-Opcode | 400005 | 00000 Hex
Quellregister-Offset | 400006 | 00000 Dez
Zielregister | 400007 | 00000 Dez
ASCII-Zeichenkette | 400008 | 00000 Dez

Zoomfenster, in denen ProWORX32 verwendet wird

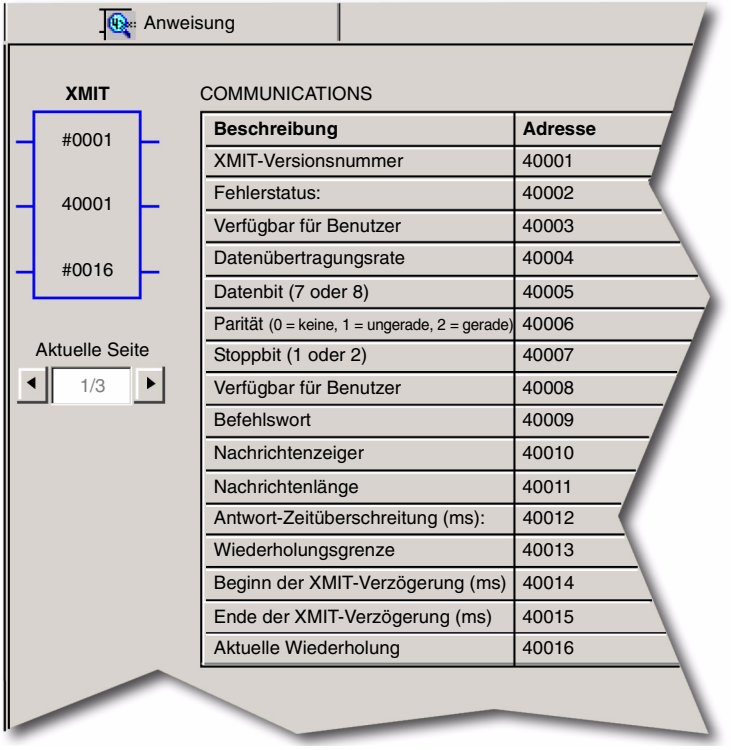
Zugriff auf die Zoomfenster mit Hilfe von ProWORX32

Ausgehend vom Ladder Logic-Editor:

Schritt	Aktion
1	Platzieren Sie einen XMIT-Block im Logikbereich.
2	Klicken Sie mit der rechten Maustaste auf den XMIT-Block.
3	Datenauswahl | Register-Editor. 

Umschalten zwischen den Zoomfenstern

Es sind drei (3) Zoomfenster verfügbar.

Schritt	Aktion
1	Wenn "Register-Editor" ausgewählt wird, wird das Zoomfenster für die Kommunikation angezeigt. 
2	Klicken Sie auf die nach links oder nach rechts gerichteten Pfeile im Feld "Aktuelle Seite". Die Zoomfenster für den Schnittstellenstatus und die Konvertierung werden in diesem Abschnitt nicht dargestellt.

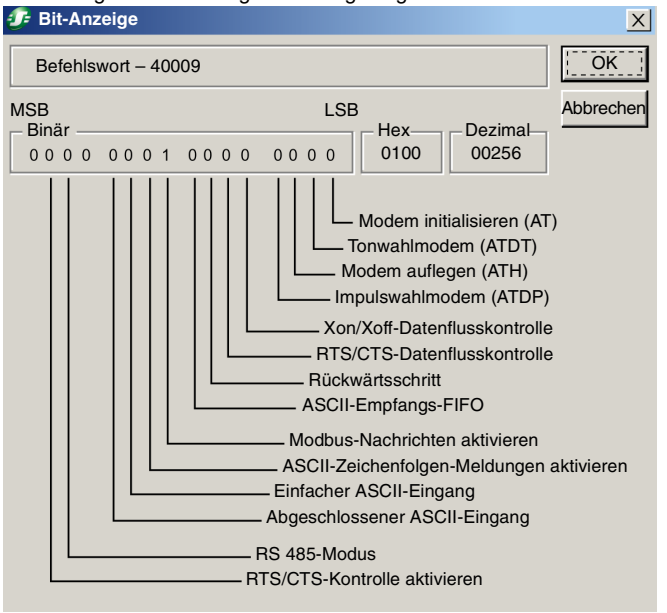
Beschreibung	Adresse
XMIT-Versionsnummer	40001
Fehlerstatus:	40002
Verfügbar für Benutzer	40003
Datenübertragungsrate	40004
Datenbit (7 oder 8)	40005
Parität (0 = keine, 1 = ungerade, 2 = gerade)	40006
Stoppbit (1 oder 2)	40007
Verfügbar für Benutzer	40008
Befehlswort	40009
Nachrichtenzeiger	40010
Nachrichtenlänge	40011
Antwort-Zeitüberschreitung (ms):	40012
Wiederholungsgrenze	40013
Beginn der XMIT-Verzögerung (ms)	40014
Ende der XMIT-Verzögerung (ms)	40015
Aktuelle Wiederholung	40016

Aufrufen der Bit-Anzeige und der Daten

Die einzelnen Bit einiger Register, wie beispielsweise "Befehlswort", aktivieren oder deaktivieren die jeweilige Funktion. Zeigen Sie den Bit-Status unter Verwendung des Dialogfelds "Bit-Anzeige" an.

Schritt	Aktion
1	Wählen Sie die ellipsenförmige Schaltfläche in der Datenspalte aus, um das Dialogfeld "Bit-Anzeige" aufzurufen.
2	Das Dialogfeld "Bit-Anzeige" wird angezeigt:

Daten
201
0
0
9600
8
2
1
0
00000001-00000000 ...
17
5
1000
5
100
100
0



Verwendung des XMIT-Funktionsblocks

4

Auf einen Blick

Einführung

In diesen Unterlagen erhalten Sie Informationen zum XMIT-Kommunikationsblock.

Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Abschnitte:

Abschnitt	Thema	Seite
4.1	Beschreibung des XMIT-Kommunikationsblocks	70
4.2	Verwendung der XMIT-Kommunikationsblockregister	78
4.3	Beschreibung und Verwendung des XMIT-Schnittstellenstatusblocks	104
4.4	Beschreibung des XMIT-Konvertierungsblocks	112
4.5	Verwendung des XMIT-Konvertierungsblocks	117
4.6	Beispiele für den XMIT-Konvertierungsblock-Opcode	123

4.1 Beschreibung des XMIT-Kommunikationsblocks

Auf einen Blick

Zweck

In diesem Abschnitt wird die Kompatibilität des XMIT-Kommunikationsblocks mit Produkten von Schneider Electric beschrieben. Darüber hinaus erhalten Sie Informationen zu XMIT-Funktionen.

Inhalt dieses Abschnitts

Dieser Abschnitt enthält die folgenden Themen:

Thema	Seite
XMIT- und SPS-Kompatibilität	71
Struktur des XMIT-Funktionsblocks	73
Inhalt des XMIT-Eintrags	75
XMIT-Kommunikationsfunktionen	77

XMIT- und SPS-Kompatibilität

SPS-Kompatibilität

Der XMIT-Funktionsblock ist mit folgenden SPS von Schneider Electric kompatibel.

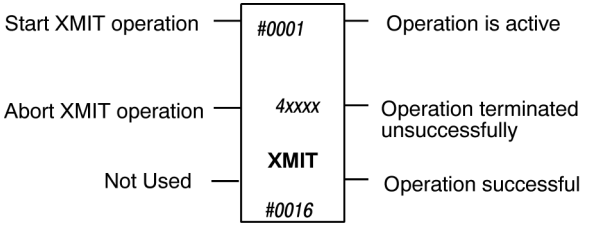
Produktfamilie	Modellnummern
Quantum PLCs	140CPU11302
	140CPU21304
	140CPU11303 (mit Hauptsteuerprogramm 2.12 oder höher)
	140CPU42402 (mit Hauptsteuerprogramm 2.10 oder höher)
	140CPU43412
	140CPU53414 (mit Hauptsteuerprogramm 1.02 oder höher)
	140CPU43412A
	140CPU53514A
Compact-SPS	PC E984 241
	PC E984 245
	PC E984 251
	PC E984 255 (mit Hauptsteuerprogramm 1.02 oder höher)
	984-E285
	984-E265
	984-E275
	984-285
Micro PLCs	110CPU61204 (mit Hauptsteuerprogramm 1.00 oder höher)

Produktfamilie	Modellnummern
Momentum-SPS	171CCS70000
	171CCS70010
	171CCS76000
	171CCS78000
	171CCC76010
	171CCC78010 (mit Hauptsteuerprogramm 2.00 oder höher)
	171CCC98020
	171CCC96020
	171CCC98030
	171CCC96030
	171CCC98091
	171CCC96091
Momentum-Steuerungen unterstützen nur ein Stoppbit.	

Struktur des XMIT-Funktionsblocks

Darstellung

Die Höhe des XMIT-Kommunikationsblocks beträgt drei Einträge.



Eingänge

XMIT verfügt über zwei mögliche Steuerungseingänge.
Der Eingang für den oberen Knoten startet einen XMIT-Vorgang und sollte so lange auf **ON** bleiben, bis der Vorgang erfolgreich abgeschlossen wurde oder ein Fehler aufgetreten ist.
Der Eingang des mittleren Knotens bricht alle aktiven XMIT-Vorgänge ab und versetzt die Schnittstelle in den Slave-Modus. Ein Abbruchcode (121) wird in das Fehlerstatusregister geschrieben. Die Schnittstelle bleibt so lange geschlossen, bis der Eingang auf ON gesetzt ist.

Hinweis: Um einen XMIT-Fehler zurückzusetzen und das Fehlerregister zu löschen, muss der obere Eingang für mindestens einen SPS-Programmzyklus auf OFF geschaltet werden.

Ausgänge

XMIT kann drei mögliche Ausgänge erstellen.
Die Ausgänge vom oberen Eintrag wechseln in den Zustand ON, wenn ein XMIT-Vorgang ausgeführt wird.
Der Ausgang vom mittleren Eintrag wird auf ON geschaltet, wenn XMIT einen Fehler erkannt hat oder ein Abbruchbefehl gegeben wurde.
Der Ausgang vom unteren Eintrag wechselt in den Zustand ON, wenn ein XMIT-Vorgang erfolgreich abgeschlossen wurde.
Die folgenden Hinweise gelten NUR FÜR LOADABLES.

Hinweis: AUSGÄNGE EIN
Alle drei Ausgänge werden, unabhängig von den Eingangszuständen, eingeschaltet, wenn für das NSUP-Loadable Folgendes gilt:

- Es wurde NICHT installiert
- Es wurde NACH dem XMIT-Loadable installiert
- Es wurde auf einer Quantum-SPS mit einer älteren Hauptsteuerungsprogramm-Version installiert als angegeben (siehe *SPS-Kompatibilität, S. 71*)

Hinweis: NSUP.EXE VOR XMIT.EXE LADEN
Die Datei `NSUP.EXE` MUSS VOR der Datei `XMIT.EXE` auf der SPS geladen werden. Ist dies nicht der Fall, so funktioniert die XMIT-Anweisung nicht ordnungsgemäß, und alle drei Ausgänge werden eingeschaltet.

Inhalt des XMIT-Eintrags

Inhalt des oberen Eintrags

Der obere Eintrag muss eine der beiden folgenden Konstanten enthalten:
- (#0001) zur Auswahl von SPS-Schnittstelle Nr. 1
- (#0002) zur Auswahl von SPS-Schnittstelle Nr. 2

WICHTIG:
- LOADABLE AKZEPTIERT 4x-Register im oberen Eintrag.
- INTEGRATION AKZEPTIERT KEINE 4x-Register im oberen Eintrag.

Inhalt des mittleren Eintrags

Das im mittleren Eintrag eingegebene 4x-Register ist das erste in einer Gruppe von sechzehn (16) zusammenhängenden Ausgangsregistern, die den Steuerungsblock, wie in der nachfolgenden Tabelle dargestellt, beinhalten:
Beschreibung der XMIT-Tabelle für die Kommunikationssteuerung.

Register	Beschreibung	Gültige Einträge
4x	XMIT-Versionsnummer	Nur lesen
4x + 1	Fehlerstatus	Nur lesen
4x + 2	Verfügbar für Benutzer	0 Kann als Zeiger für Anweisungen, wie beispielsweise TBLK, verwendet werden
4x + 3	Datenübertragungsrate	50, 75, 110, 134, 150, 300, 600, 1200, 2400, 9600 und 19200
4x + 4	Datenbits	7, 8
4x + 5	Parität	0, 1, 2
4x + 6	Stoppbit	0, 1, 2
4x + 7	Verfügbar für Benutzer	0 Kann als Zeiger für Anweisungen, wie beispielsweise TBLK, verwendet werden
4x + 8	Befehlswort	0000-0000-0000-0000
4x + 9	Zeiger auf Meldungstabelle	Begrenzt durch den Bereich des konfigurierten 4x-Registers
4x + 10	Nachrichtenlänge	0 bis 1024 (für ASCII-Meldungen) Für Modbus-Meldungen siehe *Grenzwerte für Modbus-Anfrage-/Antwortparameter, S. 178*.
4x + 11	Antwort-Zeitüberschreitung (ms)	0 bis 65535
4x + 12	Wiederholungsgrenze	0 bis 65535
4x + 13	Beginn der Übertragungsverzögerung (ms)	0 bis 65535
4x + 14	Ende der Übertragungsverzögerung (ms)	0 bis 65535
4x + 15	Aktuelle Wiederholung	Nur lesen

Hinweis: ADRESSE NICHT ÄNDERN
Die Adresse im mittleren Eintrag darf NICHT geändert oder gelöscht werden, während das Programm aktiv ist. Anderenfalls wird die Schnittstelle blockiert, und die Kommunikation wird unterbrochen.

Inhalt des unteren Eintrags

Der untere Eintrag muss eine Konstante enthalten, die (#0016) entspricht. Diese Konstante ist die Anzahl der Register, die durch die XMIT-Anweisung verwendet wird.

XMIT-Kommunikationsfunktionen

Funktionen des XMIT-Kommunikations-blocks

Der XMIT-Kommunikationsblock führt die nachfolgend aufgeführten Funktionen aus. Für jede Funktion müssen bestimmte Bits des Befehlsworts (4x + 8) eingestellt werden.
Siehe hierzu die Zusammenfassungstabelle für die Befehlswortfunktionen 4x + 8. (Siehe *(4x + 8) Bit-Zusammenfassung, S. 83*)

(4x + 8) Befehlswortfunktion	**Befehlswortbits, die auf 1 gesetzt werden müssen**	**Bits, die auf 0 gesetzt werden MÜSSEN**
Abgeschlossener ASCII-Eingang (Bit 5=1) *****	9	6, 7, 8, 13, 14, 15, 16
Einfacher ASCII-Eingang (Bit 6=1) *****	9	5, 7, 8, 13, 14, 15, 16
Modemausgang (Bit 7=1)	2, 3, 13, 14, 15, 16	5, 6, 8, 9, 10, 11, 12 (plus eins, jedoch NUR eines der folgenden Bits ist auf 1 gesetzt: 13, 14, 15 oder 16, wobei die anderen drei Bits auf 0 gesetzt werden müssen)
NUR ASCII-Empfangs-FIFO aktivieren (Bit 9=1)	2, 3, 10, 11, 12	5, 6, 7, 8, 13, 14, 15, 16
***** Wenn Sie eine dieser Funktionen verwenden, MÜSSEN Sie "ASCII-Empfangs-FIFO aktivieren" (4x + 8, Bit 9) auf 1 setzen.		

4.2 Verwendung der XMIT-Kommunikationsblockregister

Auf einen Blick

Zweck

In diesem Abschnitt werden die Register 4x bis 4x + 15 des Kommunikationsblocks beschrieben.

Inhalt dieses Abschnitts

Dieser Abschnitt enthält die folgenden Themen:

Thema	Seite
XMIT-Kommunikationsblock-Register 4x bis 4x + 7	79
XMIT-Kommunikationsblock, Register 4x + 8	82
XMIT-Kommunikationsblock-Register 4x + 8, Bit 5	86
XMIT-Kommunikationsblock-Register 4x + 8, Bit 6	89
XMIT-Kommunikationsblock-Register 4x + 8, Bit 9	90
XMIT-Kommunikationsblock-Register 4x + 8, Bit 10	91
XMIT-Kommunikationsblock-Register 4x + 8, Bit 11	92
XMIT-Kommunikationsblock-Register 4x + 8, Bit 12	94
XMIT-Kommunikationsblock-Register 4x + 9, Funktionscodes 01 bis 06, 15 und 16	96
XMIT-Kommunikationsblock-Register 4x + 9, Funktionscode 8	98
XMIT-Kommunikationsblock-Register 4x + 9, Funktionscodes 20 und 21	100
XMIT-Kommunikationsblock-Register 4x + 10 bis 4x + 15	102

XMIT-Kommunikationsblock-Register 4x bis 4x + 7

(4x) XMIT-Versionsnummer – Schreibgeschützt

Zeigt die aktuelle Versionsnummer des XMIT-Blocks an. Die Versionsnummer wird automatisch vom Block geladen, und der Block überschreibt alle anderen in dieses Register eingegebenen Versionsnummern.

(4x + 1) Kommunikationsfehlercode – Schreibgeschützt

Dieses Register zeigt einen vom XMIT-Block generierten Fehlercode an. Die folgende Tabelle enthält eine vollständige Liste der Fehlercodes.
Tabelle der Fehlerbeschreibungen für das Register (4x + 1).

Fehlercode	Fehlerbeschreibung
1	Modbus-Ausnahme – Ungültige Funktion
2	Modbus-Ausnahme – Ungültige Datenadresse
3	Modbus-Ausnahme – Ungültiger Datenwert
4	Modbus-Ausnahme – Slave-Gerätefehler
5	Modbus-Ausnahme – Quittieren
6	Modbus-Ausnahme – Slave-Gerät beschäftigt
7	Modbus-Ausnahme – Negatives Quittieren
8	Modbus-Ausnahme – Speicherparitätsfehler
9 ... 99	Reserviert
100	Slave-SPS-Datenbereich kann nicht gleich 0 sein
101	Master-SPS-Datenbereich kann nicht gleich 0 sein
102	Spule (0x) nicht konfiguriert
103	Halteregisterbereich (4x) nicht konfiguriert
104	Datenlänge kann nicht gleich 0 sein
105	Zeiger auf Meldungstabelle darf nicht gleich 0 sein
106	Zeiger auf Meldungstabelle liegt außerhalb des Bereichs des konfigurierten Halteregisters (4x)
107	Meldungs-Timeout übertragen Dieser Fehler wird generiert, wenn der UART-Chip eine Datenübertragung nicht innerhalb von 10 Sekunden abschließen kann. Hinweis: Dieser Fehler umgeht den Retry-Zähler und aktiviert die Fehlerausgabe beim ersten Fehler.
108	Nicht definierter Fehler
109	Modem gab die Meldung "ERROR" aus
110	Modem gab die Meldung "NO CARRIER" aus

Fehlercode	Fehlerbeschreibung
111	Modem gab die Meldung "NO DIALTONE" aus
112	Modem gab die Meldung "BUSY" aus
113	Ungültige LRC-Prüfsumme der Slave-SPS
114	Ungültige CRC-Prüfsumme der Slave-SPS
115	Ungültiger Modbus-Funktionscode
116	Modbus-Antwortnachrichten-Timeout
117	Modem-Antwort-Timeout
118	XMIT konnte nicht auf die SPS-Kommunikationsschnittstelle Nr. 1 oder Nr. 2 zugreifen
119	XMIT konnte den Receiver der SPS-Schnittstelle nicht aktivieren
120	XMIT konnte den SPS-UART-Chip nicht einstellen
121	Benutzer hat einen Abbruchbefehl gegeben
122	Der obere Eintrag von XMIT ist nicht gleich null, eins oder zwei
123	Der untere Eintrag von XMIT ist nicht gleich sieben, acht oder sechzehn
124	Nicht definierter interner Status
125	Der Broadcast-Modus ist bei diesem Modbus-Funktionscode nicht zulässig.
126	DCE aktivierte CTS nicht
127	Unzulässige Konfiguration (Baudrate, Datenbits, Parität oder Stoppbits)
128	Unerwartete Antwort vom Modbus-Slave empfangen
129	Unzulässige Befehlsworteinstellung
130	Befehlswort im aktiven Zustand geändert
131	Ungültige Zeichenzählung
132	Ungültiger Registerblock
133	FIFO-Überschreitungsfehler am ASCII-Eingang
134	Ungültige Anzahl von Start- oder Endzeichen

(4x + 2) Verfügbar für Benutzer

Der XMIT-Block verwendet dieses Register nicht. Es wird jedoch unter Umständen als Zeiger in der Ladder Logic verwendet. Der XMIT-Block lässt sich effizient verwenden, indem ein Zeigerwert einer TBLK-Anweisung in diesem Register platziert wird.

Datenübertragungsrate (4x + 3)

XMIT unterstützt folgende Datenübertragungsraten: 50, 75, 110, 134, 150, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 7200, 9600 und 19200. Um eine Datenübertragungsrate zu konfigurieren, geben Sie dessen Dezimalzahl in dieses Feld ein. Sobald eine ungültige Baudrate eingegeben wird, zeigt der Block einen Fehler bedingt durch eine unzulässige Konfiguration (Fehlercode 127) im Fehlerstatusregister (4x + 1) an.

Datenbits (4x + 4)

XMIT unterstützt folgende Datenbits: 7 und 8. Um ein Datenbit zu konfigurieren, geben Sie dessen Dezimalzahl in dieses Register ein. Modbus-Nachrichten können im ASCII- oder RTU-Modus verschickt werden. Der ASCII-Modus erfordert 7 Datenbits, der RTU-Modus 8 Datenbits. Beim Versenden von Nachrichten mit ASCII-Zeichen können Sie entweder 7 oder 8 Datenbits verwenden. Sobald eine ungültige Datenübertragungsrate eingegeben wird, zeigt der Block einen Fehler bedingt durch eine unzulässige Konfiguration (Fehlercode 127) im Fehlerstatusregister (4x + 1) an. Weitere Informationen zu den Modbus-Nachrichtenformaten erhalten Sie im Referenzhandbuch für Modicon Modbus-Protokolle (PI-MBUS-300).

(4x + 5) Paritätsbits

XMIT unterstützt folgende Parität: keine, ungerade oder gerade.
Geben Sie für jede der folgenden Kombinationen jeweils eine Dezimalzahl ein:

- 0 = Keine Parität
- 1 = Ungerade Parität
- 2 = Gerade Parität

Sobald eine ungültige Parität eingegeben wird, zeigt der Block einen Fehler bedingt durch eine unzulässige Konfiguration (Fehlercode 127) im Fehlerstatusregister (4x + 1) an.

(4x + 6) Stoppbits

XMIT unterstützt ein oder zwei Stoppbits.
Geben Sie für jede der folgenden Kombinationen jeweils eine Dezimalzahl ein:

- 1 = Ein Stoppbit
- 2 = Zwei Stoppbits

Sobald ein ungültiges Stoppbit eingegeben wird, zeigt der Block einen Fehler bedingt durch eine unzulässige Konfiguration (Fehlercode 127) im Fehlerstatusregister (4x + 1) an.

(4x + 7) Verfügbar für Benutzer

Der XMIT-Block verwendet dieses Register nicht. Es wird jedoch unter Umständen als Zeiger in der Ladder Logic verwendet. Der XMIT-Block lässt sich effizient verwenden, indem ein Zeigerwert einer TBLK-Anweisung in diesem Register platziert wird.

XMIT-Kommunikationsblock, Register 4x + 8

Überblick

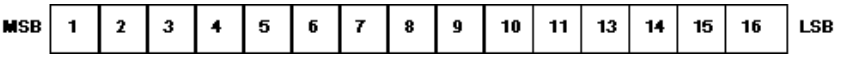
In diesem Abschnitt werden das Befehlswortregister 4x + 8 und die 16 Bit in diesem Register beschrieben.

Sechs Bit des Registers 4x + 8 werden detailliert beschrieben.

- Bit 5: Funktion Abgeschlossener ASCII-Eingang
- Bit 6: Funktion Einfacher ASCII-Eingang
- Bit 9: Funktion Einfacher ASCII-Eingang
- Bit 10: Rückwärtsschritt aktivieren
- Bit 11: RTS/CTS-Datenflusskontrolle aktivieren
- Bit 12: Xon/Xoff -Datenflusskontrolle aktivieren

(4x + 8) Befehlswort

XMIT interpretiert jedes Bit des Befehlsworts als auszuführende Funktion. Wenn Bit 7 und 8 gleichzeitig aktiv sind oder wenn jeweils 2 oder mehr der Bits 13, 14, 15 oder 16 aktiv sind oder wenn Bit 7 nicht aktiv ist, sobald die Bits 13, 14, 15 oder 16 aktiv sind, wird Fehler 129 generiert. Es gelten weitere Einschränkungen. Die einzelnen Bitdefinitionen finden Sie in der nachstehenden Tabelle.
Bit-Verteilung



(4x + 8) Bit-Zusammenfassung

Bit 1 bis 16

Bit	Beschreibung	Bit	Beschreibung
1 MSB	Reserviert für Systemverwendung.	9	ASCII-Empfangs-FIFO aktivieren
2	RTS/CTS-Modemkontrolle aktivieren	10	Rückwärtsschritt aktivieren
3	RS485-Modus aktivieren	11	RTS/CTS-Datenflusskontrolle aktivieren
4	Reserviert für Systemverwendung.	12	Xon/Xoff -Datenflusskontrolle aktivieren
5	Abgeschlossener ASCII-Eingang	13	Impulswahlmodem
6	Einfacher ASCII-Eingang	14	Modem auflegen
7	ASCII-Zeichenfolgen-Meldungen aktivieren	15	Tonwahlmodem
8	Modbus-Nachrichten aktivieren	16 LSB	Modem initialisieren

(4x + 8) Bit-Beschreibung

Bit-Definitionstabelle für das Befehlswortregister (4x + 8).

Bit	Beschreibung
Bit 1 (MSB)	**Reserviert für Systemverwendung.**
Bit 2	**RTS/CTS-Modemkontrolle aktivieren** Auf Wert 1 gesetzt, wenn ein an die SPS angeschlossenes DCE einen Hardware-Handshake mittels der RTS/CTS-Steuerung erfordert. Dieses Bit kann zusammen mit in (4x + 13) und (4x + 14) enthaltenen Werten verwendet werden. Beim Start der Übertragungsverzögerung wird RTS (4x + 13) weiterhin für (X ms) aktiviert, bevor XMIT eine Nachricht von der SPS-Schnittstelle Nr. 1 absendet. Genauso bleibt beim Ende der Übertragungsverzögerung (4x + 14) RTS weiterhin für (X ms) aktiviert, nachdem XMIT das Senden einer Nachricht von der SPS-Schnittstelle Nr. 1 beendet hat. Sobald das Ende der Übertragungsverzögerung abgelaufen ist, wird RTS durch XMIT nicht mehr länger aktiviert.
Bit 3	**RS485-Modus aktivieren** Auf Wert 1 gesetzt, wenn die ausgewählte Schnittstelle im RS485-Modus arbeiten soll. Ansonsten wird mit dem Standardwert 0 der RS232-Modus verwendet.
Bit 4	**Reserviert für Systemverwendung.**
Bit 5	**Abgeschlossener ASCII-Eingang** Wählen Sie den Wert 1, um alle Zeichen aus dem FIFO-Chip zu entfernen bzw. zu löschen, bis die Startzeichenkette gefunden wird. Dann werden diese Startzeichen sowie alle folgenden Zeichen in einen benachbarten 4x-Registerblock geschrieben, bis die Endsequenz gefunden wird. Die Endsequenz wird ebenfalls in den benachbarten 4x-Registerblock geschrieben.

Bit	Beschreibung
Bit 6	**Einfacher ASCII-Eingang** Auf Wert 1 gesetzt, um die ASCII-Zeichen aus dem FIFO-Chip zu entfernen, damit in einen benachbarten 4x-Registerblock geschrieben werden kann. Der Nachrichtenzeiger (4x + 9) gibt den 4x-Registerblock an.
Bit 7	**ASCII-Zeichenfolgen-Meldungen aktivieren** Wählen Sie den Wert 1, wenn Sie ASCII-Nachrichten von der SPS aus versenden möchten. XMIT kann ASCII-Zeichenketten mit einer Zeichenkettenlänge bis zu 1024 versenden. Sie programmieren jede ASCII-Meldung in benachbarten 4x-Registern der SPS. Pro Register sind zwei Zeichen zulässig. Verwenden Sie nur Bit 7 ODER Bit 8, jedoch keinesfalls beide gleichzeitig.
Bit 8	**Modbus-Nachrichten aktivieren** Wählen Sie den Wert 1, wenn Sie Modbus-Nachrichten von der SPS aus versenden möchten. Modbus-Nachrichten können entweder im RTU- oder ASCII-Format vorliegen. Wenn Datenbit = 8, verwendet XMIT das Modbus-RTU-Format. Wenn Datenbit = 7, verwendet XMIT das Modbus-ASCII-Format. Verwenden Sie nur Bit 7 ODER Bit 8, jedoch keinesfalls beide gleichzeitig.
Bit 9	**ASCII-Empfangs-FIFO aktivieren** Wählen Sie 1, damit der XMIT-Block die gewählte Schnittstelle (1 oder 2) der SPS überwachen kann. Der Block beginnt damit, ASCII-Zeichen in einem leeren Ring-FIFO von 512 K Größe zu empfangen.
Bit 10	**Rückwärtsschritt aktivieren** Setzen Sie diesen Wert auf 1, um eine spezielle Handhabung des ASCII-Rückwärtsschritts zu ermöglichen (BS, 8Hex). Wenn Sie entweder einen einfachen ASCII-Eingang (Bit 6) oder einen abgeschlossenen ASCII-Eingang (Bit 5) verwenden, wird jedes Rückwärtsschritt-Zeichen vom FIFO entfernt und wird in einem 4x-Zielregisterblock gespeichert oder NICHT.
Bit 11	**RTS/CTS-Datenflusskontrolle aktivieren** Wählen Sie 1, um eine Hardwareflusssteuerung bei Verwendung von RTS- und CTS-Handshake-Signalen zur ASCII-Nachrichtenübermittlung zu ermöglichen. RTS/CTS arbeiten sowohl im Eingangs- als auch Ausgangsmodus.
Bit 12	**Xon/Xoff -Datenflusskontrolle aktivieren** Wählen Sie 1, um eine Softwareflusssteuerung bei Verwendung von ASCII-Xon-Zeichen (DC1, 11 Hex) und ASCII-Xoff-Zeichen (DC3, 13 Hex) zu ermöglichen. Xon/Xoff arbeiten sowohl im Eingangs- als auch Ausgangsmodus.

Bit	Beschreibung
Bit 13	**Impulswahlmodem** Wählen Sie 1, wenn Sie ein Hayes-kompatibles Impulswahlmodem verwenden und Sie durch Impulswahl eine Telefonnummer anwählen möchten. Sie programmieren die Telefonnummer in benachbarten 4x-Registern der SPS. Ein Zeiger auf diese Register muss im Steuerungstabellenregister (4x + 9) und die Länge der Meldung in (4x + 10) gespeichert werden. Impulswahlnummern werden an das Modem gesendet, wobei automatisch ATDP und Wagenrücklauf <CR> sowie Zeilenvorschub <LF> angehängt werden. Da es sich bei der Wählnachricht um eine ASCII-Zeichenkette handelt, muss Bit 7 AKTIV sein, bevor die anzuwählende Nummer gesendet werden kann.
Bit 14	**Modem auflegen** Wählen Sie 1, wenn Sie ein Hayes-kompatibles Impulswahlmodem verwenden und die Modemverbindung trennen möchten. Sie müssen die Ladder Logic verwenden, um dieses Bit zu aktivieren. Da es sich bei der Auflegenachricht um eine ASCII-Zeichenkette handelt, muss Bit 7 AKTIV sein, bevor die Nachricht gesendet werden kann. Alle Nachrichten werden an das Modem gesendet, wobei automatisch +++AT und Wagenrücklauf <CR> sowie Zeilenvorschub <LF> angehängt werden. XMIT sucht nach einer korrekten Antwort des Modems auf das Trennen der Verbindung, bevor es den unteren Ausgang aktiviert und einen erfolgreichen Abschluss registriert.
Bit 15	**Tonwahlmodem** Wählen Sie 1, wenn Sie ein Hayes-kompatibles Impulswahlmodem verwenden und Sie durch Tonwahl eine Telefonnummer anwählen möchten. Sie programmieren die Wählnachricht in benachbarten 4x-Registern der SPS. Ein Zeiger auf die Wählnachricht muss im Steuerungstabellenregister (4x + 9) und die Länge der Meldung in (4x + 10) gespeichert werden. Tonwahlnummern werden an das Modem gesendet, wobei automatisch ATDT und Wagenrücklauf <CR> sowie Zeilenvorschub <LF> angehängt werden. Da es sich bei der Wählnachricht um eine ASCII-Zeichenkette handelt, muss Bit 7 AKTIV sein, bevor die anzuwählende Nummer gesendet werden kann.
Bit 16 (LSB)	**Modem initialisieren** Wählen Sie 1, wenn Sie ein Hayes-kompatibles Impulswahlmodem verwenden und das Modem initialisieren möchten. Sie programmieren die Initialisierungsnachricht in benachbarten 4x-Registern der SPS. Ein Zeiger auf die Initialisierungsnachricht muss im Steuerungstabellenregister (4x + 9) und die Länge der Meldung in (4x + 10) gespeichert werden. Alle Nachrichten werden an das Modem gesendet, wobei automatisch AT und Wagenrücklauf <CR> sowie Zeilenvorschub <LF> angehängt werden. Da es sich bei der Initialisierungsnachricht um eine ASCII-Zeichenkette handelt, muss Bit 7 AKTIV sein, bevor die Nachricht gesendet werden kann.

XMIT-Kommunikationsblock-Register 4x + 8, Bit 5

Funktion Abgeschlossener ASCII-Eingang (4x + 8, Bit 5)

Wenn (4x + 8, Bit 5) für abgeschlossene ASCII-Eingangsmeldungen aktiviert ist, ist der Nachrichtenzeiger (4x + 9) das Register-Offset für das erste Register der ASCII-Eingangs-Definitionstabelle. Die Definitionstabelle des abgeschlossenen ASCII-Eingangs umfasst fünf Register. Aus diesem Grund muss das Register für die Nachrichtenlänge (4x + 10) auf fünf gesetzt werden, damit XMIT ordnungsgemäß funktioniert. Die folgende Tabelle enthält die Definitionstabelle des abgeschlossenen ASCII-Eingangs. Geben Sie Ihre Daten in die Definitionstabelle für den ASCII-Eingang ein, und verwenden Sie dabei den Referenzabschnitt in Modsoft.

Definitionstabelle des abgeschlossenen ASCII-Eingangs.

Wort	Höherwertiges Byte	Niederwertiges Byte
4x +0	Anzahl der Startzeichen (zulässige Werte = 0, 1, 2)	Anzahl der Endzeichen (zulässige Werte = 1, 2)
4x +1	Erstes Startzeichen	Zweites Startzeichen
4x + 2	Erstes Endzeichen	Zweites Endzeichen
4x +3	Erstes 4x-Speicherzielregister	
4x +4	Zähler Zählt die empfangenen Zeichen, die in die 4x-Speicherzielregister geschrieben werden	

Während des Prozesses enthält das Register (4x + 4) der Definitionstabelle für den ASCII-Eingang einen laufenden Zähler der Zeichen, die in den 4x-Zielregisterblock geschrieben wurden. Sobald die abgeschlossene Zeichenkette empfangen wird, wird der untere Ausgang des XMIT-Blocks auf ON geschaltet, und das Register (4x + 4) der Definitionstabelle für den ASCII-Eingang enthält die Gesamtlänge der empfangenen Zeichenkette, einschließlich der Start- und Endzeichenketten. An diesem Punkt verfügt der XMIT-Block immer noch über die Schnittstelle und speichert weiterhin neu empfangene Zeichen im Befehlswort des ASCII-Empfangs-FIFO, weil das Befehlsworts "ASCII-Empfangs-FIFO aktivieren" (4x + 8, Bit 9) auf ON geschaltet ist.

Mit der Ladder Logic können Sie das Bit für den einfachen ASCII-Eingang (4x + 8,Bit 6) vor dem nächsten Zyklus löschen, während das Bit für die Aktivierung des ASCII-Empfangs-FIFO (4x + 8, Bit 9) eingeschaltet bleibt. Dadurch wird der aktuelle 4x-Registerzielblock NICHT durch neuere FIFO-Daten überschrieben, die immer noch im FIFO gesammelt werden. Mit der Ladder Logic können Sie beide Bit zum Aktivieren des ASCII-Empfangs-FIFO (4x + 8, Bit 9) und des abgeschlossenen ASCII-Eingangs (4x + 8, Bit 5) löschen, um die Schnittstellensteuerung der SPS zu überlassen.

Wenn OHNE erkanntes Endzeichen zu viele Zeichen in den 4x-Registerzielblock geschrieben werden oder die Matrix des 4x-Registerzielblocks außerhalb des zulässigen Bereichs für die konfigurierte SPS liegt, wird im Fehlerstatus (4x +1) ein Fehler gemeldet. Die Zeichenbegrenzung liegt bei maximal 1024 Zeichen oder bei der doppelten Größe des 4x-Registerzielblocks.
Es empfiehlt sich, den 4x-Registerzielblock für den abgeschlossenen ASCII-Eingang (4x + 8,Bit 5) nach allen anderen 4x-Registern zu platzieren, die Sie in der Anwendung verwenden, um zu vermeiden, dass sie durch den ASCII-Eingang überschrieben werden, falls der Terminator nicht vorhanden ist. Darüber hinaus können Sie 512 Register für den 4x-Registerzielblock zuweisen.

Abgeschlossener ASCII-Eingang (4x + 8, Bit 5) – Beispiel

Angenommen, XMIT wurde durch Setzen des Befehlsworts (4x + 8, Bit 9 und 5) aktiviert. Aktivieren Sie das ASCII-FIFO und das abgeschlossene ASCII. Die folgende ASCII-Zeichenkette wird von der Schnittstelle empfangen: "AMScrlf$weight= 1245 GRAMScrlf$wei". Verwenden Sie die nachfolgende Definitionstabelle des ASCII-Eingangs als Referenz. In der Tabelle wird der Inhalt der Register dargestellt, die durch Klammern ( ) gekennzeichnet sind.
Definitionstabelle des abgeschlossenen ASCII-Eingangs, Beispiel (Inhalt in Klammern)

Wort	Höherwertiges Byte	Niederwertiges Byte
4x +0	Anzahl der Startzeichen (0x01)	Anzahl der Endzeichen (0x02)
4x +1	Erstes Startzeichen (’$’)	Zweites Startzeichen (nicht verwendet)
4x + 2	Erstes Endzeichen (’cr’)	Zweites Endzeichen ("if")
4x +3	Erstes 4x-Speicherzielregister (101) = 400101	
4x +4	Zähler Zählt die empfangenen Zeichen, die in die 4x-Speicherzielregister (??????) geschrieben werden	

Der XMIT-Block wird AKTIV und löscht die ersten fünf Zeichen "AMScrlf" aus dem Eingangs-FIFO, da sie dem ersten Startzeichen, das auf '$' gesetzt ist, nicht entsprechen. Im Zyklus nach dem Empfang des Zeichens '$' bleibt der XMIT-Block AKTIV und kopiert das Zeichen '$' und die folgenden Zeichen in den 4x-Zielspeicher, wobei (4x + 4) der Definitionstabelle für den ASCII-Eingang mit der bisherigen Anzahl aktualisiert wird, wenn Zeichen eingehen. Nach Empfang des letzten Endzeichens wird "Vorgang erfolgreich" am unteren Ausgang aktiviert, und (4x + 4) der Definitionstabelle für den ASCII-Eingang enthält die Gesamtlänge 0x0016. Der 4x-Zielspeicherblock beginnt bei 400101 und enthält Folgendes: "$w", "ei", "gh", "t", "=", "12", "45", "G", "RA", "MS", "cflf". In dem Zyklus, in dem der "Vorgang erfolgreich" am unteren Ausgang aktiviert wird, verbleiben die bereits empfangenen Zeichen aus der nächsten Meldung, "$wei", die nach der Endzeichenkette eintrafen, weiterhin im ASCII-Eingangs-FIFO. Dadurch erhält die Ladder Logic die Gelegenheit, den Eingang für abgeschlossenes ASCII (4x + 8, Bit 5) abzuschalten, bevor der nächste Abtastzyklus von XMIT für diese Schnittstelle erfolgt. So bleiben diese Zeichen im FIFO, bis die SPS die Verarbeitung der aktuellen Meldung abgeschlossen hat, wozu mehrere Zyklen erforderlich sein können.

XMIT-Kommunikationsblock-Register 4x + 8, Bit 6

(4x + 8, Bit 6) Einfache ASCII-Eingangs-funktion

In jedem 4x-Register sind zwei Zeichen gespeichert. Das erste Zeichen, das vom FIFO übertragen wird, wird im höherwertigen Byte des ersten 4x-Registers gespeichert. Das zweite Zeichen, das vom FIFO übertragen wird, wird im höherwertigen Byte des ersten 4x-Registers gespeichert. Das dritte Zeichen wird im höherwertigen Byte des zweiten 4x-Registers gespeichert usw. Das Register für die Nachrichtenlänge (4x + 10) enthält die Länge der Nachricht (1 bis 1024). Daher nimmt das Register für die Nachrichtenlänge (4x + 10) ab, wenn die Zeichen vom FIFO in den benachbarten 4x-Registerblock übertragen werden. Sobald die gesamte Meldung übertragen wurde, stellt das Register für die Nachrichtenlänge (4x +10) seinen ursprünglichen Wert wieder her, und der Ausgang "Vorgang erfolgreich" von XMIT wird aktiviert.

Hinweis: Wenn "Einfacher ASCII-Eingang" (Bit 6) und "ASCII-Empfangs-FIFO" (Bit 9) weiterhin gesetzt sind, werden neue Zeichen fortlaufend vom FIFO in denselben 4x-Registerblock übertragen. Somit werden die zuvor im 4x-Registerblock gespeicherten Zeichen ständig neu überschrieben.

Durch Löschen des einfachen ASCII-Eingangs (Bit 7) vor dem nächsten Zyklus in Ihrer Ladder Logic und durch die Einstellung des ASCII-Empfangs-FIFO (Bit 9) können Sie weiterhin neue Zeichen aufzeichnen und das ständige Überschreiben des 4x-Registerblocks vermeiden. Wenn Sie sowohl den einfachen ASCII-Eingang (Bit 7) als auch das ASCII-Empfangs-FIFO (Bit 9) mit Hilfe der Ladder Logic löschen, wird die Steuerung der Schnittstelle (1 oder 2) an die SPS zurückgegeben. Wenn das Register für die Nachrichtenlänge (4x + 10) auf 0 oder höher als 1024 gesetzt ist oder wenn der 4x-Registerblock außerhalb des zulässigen Bereichs für die konfigurierte SPS liegt, wird ein Fehler in Fehlerstatus (4x +1) gemeldet.

XMIT-Kommunikationsblock-Register 4x + 8, Bit 9

(4x + 8, Bit 9) ASCII-Empfangs-FIFO aktivieren

Wenn dieses Bit auf 0 gesetzt wird, wird diese Funktion beendet. Sobald der FIFO-Chip 512 Zeichen empfängt, wird eine interne Überschreitung festgelegt. Sobald dies passiert, werden alle folgenden Zeichen gelöscht, alle ASCII-Eingangsoperationen (einfach und abgeschlossen) abgeschlossen, und der Block gibt einen Fehler aus, bis Bit 9 seinen Status ändert. Wenn Bit 9 seinen Status geändert hat, werden alle Daten im FIFO-Chip gelöscht und beide Steuerbit des ASCII-Eingangs ignoriert (Einfaches ASCII (Bit 6), Abgeschlossenes ASCII (Bit 5)). Wenn keine ASCII-Ausgangssteuerungen ausgewählt sind, dann wird die Steuerung der Schnittstelle (1 oder 2) wieder der SPS überlassen.
Sie müssen entweder Abgeschlossenes ASCII (Bit 5) oder Einfaches ASCII (Bit 6) eingestellt haben, um die ASCII-Zeichen vom FIFO-Chip zur Verarbeitung zu entfernen. Es kann nur 1 Bit der folgenden 3 Bit gleichzeitig gesetzt werden: Abgeschlossenes ASCII (Bit 5), Einfaches ASCII (Bit 6) oder ASCII-Ausgang (Bit 7). Vollduplexbetrieb wird erreicht, indem Sie sowohl ASCII-Empfangs-FIFO (Bit 9) als auch ASCII-Ausgang (Bit 7) einstellen. Dadurch ist eine einfache ASCII-Übertragung von der SPS weg möglich, während im FIFO-Chip immer noch ASCII-Zeichen empfangen werden. Dies ist nützlich beim Arbeiten mit Ausgabeterminals. Wenn der ASCII-Empfangs-FIFO (Bit 9) eingestellt ist, ist keine der folgenden ASCII-Ausgangsüberwachungen erlaubt: Modbus-Masternachrichten (Bit 8), Impulswahl-Modem (Bit 13), Modem auflegen (Bit 14), Tonwahl-Modem (Bit 15) und Modem initialisieren (Bit 16).

XMIT-Kommunikationsblock-Register 4x + 8, Bit 10

(4x + 8, Bit 10) Rückwärts-schritt aktivieren

Wenn ein Rückwärtsschritt (BS) erkannt wird, wird er NICHT im Zielblock des Register 4x gespeichert, sondern er löscht das vorige Zeichen und vermindert dadurch die Anzahl der abgeschlossenen (Bit 5-)Zeichen (4x + 4) in der Definitionstabelle des ASCII-Eingangs. Wenn hingegen ein normales ASCII-Zeichen erkannt wird, wird es im Zielblock des Registers 4x gespeichert, und die Anzahl der abgeschlossenen (Bit 5-)Zeichen in der Definitionstabelle des ASCII-Eingangs (4x + 4) wird erhöht.

Hinweis: Rückwärtsschritte KÖNNEN KEINE Zeichen aus einem leeren 4x-Register löschen. Daher liegt die Anzahl der abgeschlossenen (Bit 5-)Zeichen in der Definitionstabelle des ASCII-Eingangs (4x + 4) nie unter 0.

Diese spezielle Rückwärtsschrittfunktion in Verbindung mit dem am Terminal aktivierten internen Rückmeldesignal ist sehr nützlich beim Umgang mit Ausgabeterminals. Ein einzelner XMIT-Block des abgeschlossenen ASCII-Eingangs, der nach "cr" sucht, ist aktiviert, während ASCII-Empfangs-FIFO (Bit 9) und Rückwärtsschritt (Bit 10) eingestellt sind. Keine zusätzliche Ladder Logic ist erforderlich, während Sie Zeichen über Rückwärtsschritt gerade eingeben oder bearbeiten. Wenn Sie "cr" eingeben, aktiviert XMIT den unteren Ausgang "Befehl erfolgreich", und die korrigierten Daten werden alle ordnungsgemäß im Zielblock des Registers 4x aufgeführt.

XMIT-Kommunikationsblock-Register 4x + 8, Bit 11

(4x + 8, Bit 11) RTS/CTS-Datenfluss-kontrolle aktivieren

Der folgende Text bezieht sich auf den Ausgangsmodus. Die XMIT-Statusvariable wechselt in den Empfangsstatus BLOCKED, wenn das empfangende Gerät durch Setzen von CTS auf OFF anzeigt, dass es keine weiteren Zeichen verarbeiten kann. Genauso wird die XMIT-Statusvariable in den Zustand UNBLOCKED versetzt, wenn das empfangende Gerät durch Setzen von CTS auf ON anzeigt, dass es weitere Zeichen verarbeiten KANN.

Ist die Übertragung UNBLOCKED, und sind der einfache ASCII-Ausgang (Bit 7) und die Datenflusskontrolle RTS/CTS (Bit 11) gesetzt, so werden die übertragenen Ausgangsdaten in 16-Byte-Paketen gesendet. Nachdem alle ausgegebenen Pakete gesendet wurden, wechselt der untere Ausgang im XMIT-Block auf ON, um "Befehl erfolgreich" anzuzeigen.

Wenn während einer Übertragung der Zustand plötzlich in BLOCKED wechselt, werden nur die verbleibenden Zeichen im aktuellen Ausgangspaket gesendet, wobei niemals 16 Zeichen überschritten werden, und der XMIT-Block bleibt für unbegrenzte Dauer ACTIVE. Nur wenn CTS ON ist, wird der ASCII-Ausgang alle verbleibenden Ausgangspakete senden.

Der folgende Text bezieht sich auf den Eingangsmodus. Da RTS ein Ausgangssignal ist, kann es unabhängig vom ASCII-Ausgangsprozess (Senden) benutzt werden, um die sendenden Geräte zu verriegeln (BLOCK) oder zu entriegeln (UNBLOCK). Wenn der ASCII-Empfangs-FIFO (Bit 9) gesetzt wurde, arbeitet die RTS/CTS-Datenflusskontrolle im Eingangsmodus. Wenn der ASCII-Empfangs-FIFO (Bit 9) gesetzt wurde und keiner der beiden ASCII-Eingänge, der einfache ASCII-Eingang (Bit 6) oder der abgeschlossene ASCII-Eingang (Bit 5) gesetzt wurde, füllen die empfangenen Zeichen den FIFO, in den sie eingefügt wurden. In der Zwischenzeit hat die RTS-Datenflusskontrolle (Bit 11) auf ON geschaltet und ermöglicht dem sendenden Gerät, mit dem Vorgang fortzufahren.

Ist der FIFO zu mehr als drei Viertel mit Zeichen gefüllt, wird die RTS-Datenflusskontrolle (Bit 11) gelöscht, um das sendende Gerät zu sperren (BLOCK). Die RTS-Datenflusskontrolle (Bit 11) bleibt gelöscht, bis entweder der einfache ASCII-Eingang (Bit 6) oder der abgeschlossene ASCII-Eingang (Bit 5) genügend Zeichen aus dem FIFO entfernt hat, bis weniger als ein Viertel Zeichen verbleiben. An diesem Punkt wird die RTS-Datenflusskontrolle (Bit 11) eingeschaltet (ON).

Hinweis: Der Algorithmus der RTS/CTS-Datenflusskontrolle unterscheidet sich von der RTS/CTS-Modemkontrolle. Der erstgenannte bezieht sich auf einen Überlauf des Vollduplex-Empfangspuffers. Der letztgenannte arbeitet mit dem Sendeprozess, der auf ein gemeinsames Übertragungsmedium zugreift. Es ist daher unzulässig, gleichzeitig beide dieser RTS/CTS-Algorithmen aufzurufen.

Hinweis: Sie können bei der RTS/CTS-Datenflusskontrolle (Bit 11) keinen BELIEBIGEN Handshaking-Typ auswählen, wenn sich die Schnittstelle im RS 485-Modus (Bit 3) befindet, da diese Signale im RS 485-Modus NICHT vorhanden sind.

XMIT-Kommunikationsblock-Register 4x + 8, Bit 12

(4x + 8, Bit 12) Xon/Xoff-Datenfluss-steuerung aktivieren

Der folgende Text bezieht sich auf den Ausgangsmodus. Die XMIT-Statusvariable wechselt bei Empfang des Xoff-Zeichens auf "BLOCKED". Genauso wechselt die XMIT-Statusvariable bei Empfang des Xon-Zeichens auf "UNBLOCKED". Xon oder Xoff werden auf keinen Fall in den FIFO eingefügt.
Ist die Übertragung freigegeben (UNBLOCKED), und sind der einfache ASCII-Ausgang (Bit 7) und die Xon/Xoff-Datenflusskontrolle (Bit 12) gesetzt, dann werden die zu übertragenden Ausgangsdaten in 16-Byte-Paketen gesendet. Nachdem alle ausgegebenen Pakete gesendet wurden, wechselt der untere Ausgang im XMIT-Block auf ON, um "Befehl erfolgreich" anzuzeigen.
Wenn während einer Übertragung der Zustand plötzlich in BLOCKED wechselt, werden nur die verbleibenden Zeichen im aktuellen Ausgangspaket gesendet, wobei niemals 16 Zeichen überschritten werden, und der XMIT-Block bleibt für unbegrenzte Dauer ACTIVE. Nur bei Empfang des nächsten Xon-Zeichens wird der ASCII-Ausgang das Senden von allen verbleibenden Ausgangspaketen wiederholen.
Der folgende Text bezieht sich auf den Eingangsmodus. Mit Xon/Xoff können sendende Geräte gesperrt (BLOCK) oder freigegeben (UNBLOCK) werden. Wenn der ASCII-Empfangs-FIFO (Bit 9) gesetzt wurde, arbeitet die Xon/Xoff-Datenfluss-kontrolle (Bit 12) im Eingangsmodus. Wenn der ASCII-Empfangs-FIFO (Bit 9) gesetzt wurde und keiner der beiden ASCII-Eingänge, der einfache ASCII-Eingang (Bit 6) oder der abgeschlossene ASCII-Eingang (Bit 5), gesetzt wurde, füllen die empfangenen Zeichen den FIFO.
Wenn der FIFO zu mehr als drei Viertel voll mit Zeichen ist und zusätzliche Zeichen empfangen werden, wird die FIFO-Statusvariable gesetzt, um XOFF-Zeichen nach einer Verzögerung von 16 Zeichen aus von der seriellen Schnittstelle zu senden, wobei der Sender gesperrt wird (BLOCKING) und die FIFO-Statusvariable gelöscht wird.
Wenn alle ASCII-Ausgangsfunktionen (Bits 8,13,14,15 und 16) im Zustand OFF sind, und die Xon/Xoff-Datenflusskontrolle (Bit 12) ist ON, ist die Verzögerungszeit standardmäßig 1 Zeichen/Takt. Sind jedoch alle ASCII-Ausgangsfunktionen (Bits 8,13,14,15 und 16) im Zustand ON, und die Xon/Xoff-Datenflusskontrolle (Bit 12) ist ON, wird der ASCII-Ausgang in Pakete von jeweils 16 Byte unterteilt. Daher müssen anhängige Xoff-Zeichen "nicht" länger als 16 Zeichen/Takt warten, bevor der Sender gesperrt wird.
Sobald der Sender die Übertragung gestoppt hat, entfernt die Steuerung eventuell die Zeichen aus dem FIFO. Sie benutzt dazu entweder den einfachen ASCII-Eingang (Bit 6) oder den abgeschlossenen ASCII-Eingang (Bit 7).
Sind im FIFO weniger als ein Viertel Zeichen, wird die FIFO-Statusvariable gesetzt, damit XON gesendet werden kann. Dadurch wird ein Xon-Zeichen an die serielle Schnittstelle gesendet, um den Sender auf UNBLOCK zu setzen.

Hinweis: Um eine Blockierung infolge eines losen Kabels oder von sonstigen intermittierenden Kommunikationsfehlern zu verhindern, wenn der Sender gesperrt (BLOCKED) ist und das Xon-Zeichen nicht korrekt empfangen hat, wird der folgende Algorithmus benutzt. Wird der FIFO leer und werden keine weiteren Zeichen empfangen, wird ein stetiger Strom mit Xon-Zeichen mit einer Geschwindigkeit von einem Zeichen alle 5 Sekunden empfangen.

Hinweis: Die Xon/Xoff-Datenflusskontrolle (Bit 12) unterscheidet sich von der RTS/CTS-Datenflusskontrolle (Bit 11). Die Erstgenannte benutzt übertragene Xon- und Xoff-Zeichen, um einen Überlauf des Empfangspuffers im Vollduplexbetrieb zu verhindern. Die Letztgenannte benutzt Hardware-Handshaking-Signale, um dasselbe Ziel zu erreichen. Es ist deshalb unzulässig, gleichzeitig beide dieser Datenflusskontroll-Algorithmen anzufordern, da die RTS/CTS-Datenflusskontrolle (Bit 11) ein Halbduplexnetz benötigt, während die Xon/Xoff-Datenflusskontrolle (Bit 12) ein Vollduplexnetz braucht.

XMIT-Kommunikationsblock-Register 4x + 9, Funktionscodes 01 bis 06, 15 und 16

(4x + 9) Zeiger auf Meldungstabelle

Geben Sie einen Zeiger ein, der auf den Beginn der Meldungstabelle verweist. Für ASCII-Zeichenfolgen ist der Zeiger das Register, dessen Offset auf das erste Register in der ASCII-Zeichenfolge verweist. Jedes Register enthält bis zu zwei ASCII-Zeichen. Jede ASCII-Zeichenkette kann bis zu 1024 Zeichen aufweisen. Wenn Sie beispielsweise 10 ASCII-Meldungen von der SPS aus senden möchten, müssen Sie 10 ASCII-Zeichenfolgen in den 4x-Registern der SPS programmieren und anschließend mit Hilfe der Ladder Logic den Zeiger auf den Beginn der jeweiligen Meldung nach jedem erfolgreichen XMIT-Vorgang setzen.

(4x + 9) – Modbus-Funktionscodes (01 bis 06, 15 und 16) Zeiger auf Meldungstabelle

Für Modbus-Meldungen ist der Zeiger das Register, dessen Offset auf das erste Register in der Modbus-Definitionstabelle verweist. Die Modbus-Definitionstabelle für den Modbus-Funktionscode: 01, 02, 03, 04, 05, 06, 15 und 16 ist fünf Register lang, und Sie müssen ihn für eine erfolgreiche XMIT-Operation programmieren. Die Modbus-Definitionstabelle sehen Sie in der folgenden Tabelle.
Siehe Funktionscodes (01 bis 06, 15 und 16) der Modbus-Definitionstabelle.

Register	Beschreibung
4y Modbus-Funktionscode	XMIT unterstützt die folgenden Funktionscodes: 01 = Mehrere Spulen lesen (0x) 02 = Mehrere Biteingänge lesen (1x) 03 = Mehrere Halteregister lesen (4x) 04 = Mehrere Eingangsregister lesen (3x) 05 = Einzelne Spule schreiben (0x) 06 = Einzelne Halteregister schreiben (4x) 15 = Mehrere Spulen schreiben (0x) 16 = Einzelne Halteregister schreiben (4x)
4y + 1 Menge	Geben Sie die Datenmenge ein, die auf die Slave-SPS geschrieben oder von der Slave-SPS gelesen werden soll. Geben Sie beispielsweise 100 ein, um 100 Halteregister von der Slave-SPS zu lesen, oder geben Sie 32 ein, um 32 Spulen auf eine Slave-SPS zu schreiben. Die Menge ist in Abhängigkeit vom SPS-Typ begrenzt. Unter *Grenzwerte für Modbus-Anfrage-/Antwortparameter, S. 178* finden Sie alle Einzelheiten zu Grenzwerten.
4y + 2 Slave-SPS-Adresse	Geben Sie die Slave-Modbus-SPS-Adresse ein. In der Regel liegt der Bereich der Modbus-Adresse bei 1–247. Um eine Modbus-Nachricht an mehrere SPS zu senden, geben Sie für die Slave-SPS-Adresse 0 ein. Dieses wird als Broadcast-Modus bezeichnet. Der Broadcast-Modus unterstützt nur Modbus-Funktionscodes, die Daten von einer Master-SPS in eine Slave-SPS schreiben. Der Broadcast-Modus unterstützt KEINE Modbus-Funktionscodes, die Daten von Slave-SPS lesen.

Register	Beschreibung
4y + 3 Slave-SPS-Datenbereich	Für einen Lesebefehl ist der Slave-SPS-Datenbereich die Datenquelle. Für einen Schreibbefehl ist der Slave-SPS-Datenbereich das Datenziel. Wenn Sie beispielsweise Spulen (00300–00500) von einer Slave-SPS lesen möchten, geben Sie in dieses Feld 300 ein. Wenn Sie Daten von einer Master-SPS schreiben und sie in einem Register (40100) einer Slave-SPS platzieren möchten, geben Sie in dieses Feld 100 ein. In Abhängigkeit vom Modbus-Befehlstyp (Schreiben oder Lesen) muss der Quell- und Zieldatenbereich den in der nachstehenden Tabelle definierten Quell- und Zieldatenbereichen entsprechen.
4y + 4 Master-SPS-Datenbereich	Für einen Lesebefehl ist der Master-SPS-Datenbereich der Empfänger für die vom Slave zurückgesendeten Daten. Für einen Schreibbefehl ist der Master-SPS-Datenbereich die Datenquelle. Wenn Sie beispielsweise Spulen (00016–00032) in der Master-SPS auf eine Slave-SPS schreiben möchten, geben Sie in dieses Feld 16 ein. Wenn Sie die Eingangsregister (30001–30100) von einer Slave-SPS lesen und die Daten im Datenbereich der Master-SPS (40100–40199) platzieren möchten, geben Sie in dieses Feld 100 ein. In Abhängigkeit vom Modbus-Befehlstyp (Schreiben oder Lesen) muss der Quell- und Zieldatenbereich den in der nachstehenden Tabelle definierten Quell- und Zieldatenbereichen entsprechen.

Wenn Sie von der SPS aus 20 Modbus-Nachrichten senden möchten, müssen Sie 20 Modbus-Definitionstabellen programmieren und anschließend nach jeder erfolgreichen Operation von XMIT mit Hilfe der Ladder Logic den Zeiger zu jeder Definitionstabelle erhöhen, oder Sie können 20 separate XMIT-Blöcke programmieren und sie dann nacheinander über die Ladder Logic aktivieren. Siehe Tabelle zu den Quell- und Zieldatenbereichen für Funktionscodes (01 to 06, 15 und 16).

Funktionscode	Master-SPS-Datenbereich	Slave-SPS-Datenbereich
03 (Mehrere 4x lesen)	4x (Empfänger)	4x (Quelle)
04 (Mehrere 3x lesen)	4x (Empfänger)	3x (Quelle)
01 (Mehrere 0x lesen)	0x (Empfänger)	0x (Quelle)
02 (Mehrere 1x lesen)	0x (Empfänger)	1x (Quelle)
16 (Mehrere 4x schreiben)	4x (Quelle)	4x (Empfänger)
15 (Mehrere 0x schreiben)	0x (Quelle)	0x (Empfänger)
05 (Einzelne 0x schreiben)	0x (Quelle)	0x (Empfänger)
06 (Einzelne 4x schreiben)	4x (Quelle)	4x (Empfänger)

XMIT-Kommunikationsblock-Register 4x + 9, Funktionscode 8

(4x + 9) – Modbus-Funktionscode (08), Zeiger auf Meldungstabelle

Für Modbus-Meldungen ist der Zeiger das Register, dessen Offset auf das erste Register in der Modbus-Definitionstabelle verweist. Die Modbus-Definitionstabelle für den Modbus-Funktionscode: 08 ist fünf Register lang, und Sie müssen sie für einen erfolgreichen XMIT-Betrieb programmieren. Die Modbus-Definitionstabelle sehen Sie in der folgenden Tabelle.
Siehe Tabelle mit den Funktionscodes der Modbus-Definitionstabelle (08).

Register	Beschreibung
4y Modbus-Funktionscode	XMIT unterstützt den folgenden Funktionscode: 08 = Diagnose
4y + 1 Diagnosen	Geben Sie in dieses Feld den Dezimalwert des Diagnoseunterfunktionscodes ein, um die jeweilige gewünschte Diagnosefunktion auszuführen. Die folgenden Diagnoseunterfunktionen werden unterstützt: Code und Beschreibung 00 Anfragedaten zurücksenden 01 Kommunikationsoption neu starten 02 Diagnoseregister zurücksenden 03 ASCII-Eingangsabgrenzung ändern 04 Listen-Only-Modus erzwingen 05 Reserviert 06 Reserviert 07 Reserviert 08 Reserviert 09 Reserviert 10 Zähler und Diagnoseregister in 384s, 484s löschen 11 Bus-Nachrichtenzählung zurücksenden 12 Bus-Kommunikationsfehlerzähler zurücksenden 13 Bus-Ausnahmefehlerzähler zurücksenden 14 Nicht unterstützt 15 Nicht unterstützt 16 Slave-NAK-Zähler zurücksenden 17 Slave-Besetzt-Zähler zurücksenden 18 Bus-Zeichenüberlauf-Zähler zurücksenden 19 Nicht unterstützt 20 Nicht unterstützt 21 Nicht unterstützt
4y + 2 Slave-SPS-Adresse	Geben Sie die Slave-Modbus-SPS-Adresse ein. In der Regel liegt der Bereich der Modbus-Adresse bei 1 bis 247. Funktionscode 8 unterstützt Broadcast-Modus (Adresse 0) NICHT

Register	Beschreibung
4y + 3 Diagnosefunktion Datenfeldinhalt	Sie geben hier den für den Datenbereich der jeweiligen Diagnoseunterfunktion benötigten Dezimalwert ein. Für die Unterfunktionen 02, 04, 10, 11, 12, 13, 16, 17 und 18 wird dieser Wert automatisch auf null gesetzt. Für die Unterfunktionen 00, 01 und 03 geben Sie den gewünschten Datenfeldwert ein. Weitere Informationen hierzu erhalten Sie im Referenzhandbuch zum Modicon-Modbus-Protokoll (PI-MBUS-300).
4y + 4 Master-SPS-Datenbereich	Für alle Unterfunktionen ist der Master-SPS-Datenbereich der Empfänger der vom Slave zurückgesendeten Daten. Sie legen ein 4x-Register fest, das den Anfang des Datenbereichs markiert, in dem die zurückgesendeten Daten platziert werden. Um beispielsweise Daten im Master-SPS-Datenbereich zu platzieren, der mit (40100) beginnt, geben Sie in dieses Feld 100 ein. Die Unterfunktion 04 sendet KEINE Antwort zurück. Weitere Informationen hierzu erhalten Sie im Referenzhandbuch zum Modicon-Modbus-Protokoll (PI-MBUS-300).

XMIT-Kommunikationsblock-Register 4x + 9, Funktionscodes 20 und 21

(4x + 9) – Modbus-Funktionscodes (20, 21) Zeiger auf Meldungstabelle

Für Modbus-Meldungen ist der Zeiger das Register, dessen Offset auf das erste Register in der Modbus-Definitionstabelle verweist. Die Modbus-Definitionstabelle für den Modbus-Funktionscode: 20 und 21 ist sechs Register lang, und Sie müssen sie für einen erfolgreichen XMIT-Betrieb programmieren. Die Modbus-Definitionstabelle sehen Sie in der folgenden Tabelle.
Nachfolgend wird die Modbus-Definitionstabelle dargestellt

Register	Beschreibung
4y Modbus-Funktionscode	XMIT unterstützt die folgenden Funktionscodes: 20 = Allgemeine Referenz lesen (6x) 21 = Allgemeine Referenz schreiben (6x)
4y + 1 Menge	Geben Sie die Datenmenge ein, die auf die Slave-SPS geschrieben oder von der Slave-SPS gelesen werden soll. Geben Sie z.B. 100 ein, um 100 Halteregister von der Slave-SPS zu lesen, oder geben Sie 32 ein, um 32 Spulen auf eine Slave-SPS zu schreiben. Die Menge ist in Abhängigkeit vom SPS-Typ begrenzt. Im Anhang A finden Sie alle Einzelheiten zu Grenzwerten.
4y + 2 Slave-SPS-Adresse	Geben Sie die Slave-Modbus-SPS-Adresse ein. In der Regel liegt der Bereich der Modbus-Adresse bei 1 bis 247. Funktionscode 20 und 21 unterstützen Broadcast-Modus (Adresse 0) NICHT.
4y + 3 Slave-SPS-Datenbereich	Für einen Lesebefehl ist der Slave-SPS-Datenbereich die Datenquelle. Für einen Schreibbefehl ist der Slave-SPS-Datenbereich das Datenziel. Wenn Sie beispielsweise Register (600300–600399) von einer Slave-SPS lesen möchten, geben Sie in dieses Feld 300 ein. Wenn Sie Daten von einer Master-SPS schreiben und sie in einem Register (600100) einer Slave-SPS platzieren möchten, geben Sie in dieses Feld 100 ein. In Abhängigkeit vom Modbus-Befehlstyp (Schreiben oder Lesen) muss der Quell- und Zieldatenbereich den in der nachstehenden Tabelle definierten Quell- und Zieldatenbereichen entsprechen. Das unterste erweiterte Register wird als Register "Null" (600000) adressiert. Das unterste Halteregister wird als Register "Eins" (400001) adressiert.

Register	Beschreibung
4y + 4 Master-SPS-Datenbereich	Für einen Lesebefehl ist der Master-SPS-Datenbereich der Empfänger für die vom Slave zurückgesendeten Daten. Für einen Schreibbefehl ist der Master-SPS-Datenbereich die Datenquelle. Wenn Sie beispielsweise Register (40016–40032) in der Master-SPS in 6x-Register in einer Slave-SPS schreiben möchten, geben Sie in dieses Feld 16 ein. Wenn Sie 6x-Register (600001–600100) von einer Slave-SPS lesen und die Daten im Datenbereich der Master-SPS (40100–40199) platzieren möchten, geben Sie in dieses Feld 100 ein. In Abhängigkeit vom Modbus-Befehlstyp (Schreiben oder Lesen) muss der Quell- und Zieldatenbereich den in der nachstehenden Tabelle definierten Quell- und Zieldatenbereichen entsprechen. Das unterste erweiterte Register wird als Register "Null" (600000) adressiert. Das unterste Halteregister wird als Register "Eins" (400001) adressiert.
4y + 5 Dateinummer	Geben Sie die Dateinummer für die 6x-Register ein, die in (1 bis 10) geschrieben bzw. gelesen werden sollen, je nach Größe des erweiterten Registerdatenbereichs. 600001 ist 60001 Datei 1 690001 ist 60001 Datei 10, wie vom Referenzdateneditor in Modsoft angesehen.

Wenn Sie von der SPS aus 20 Modbus-Meldungen senden möchten, müssen Sie 20 Modbus-Definitionstabellen programmieren und anschließend den Zeiger auf die Definitionstabelle nach jedem erfolgreichen XMIT-Vorgang über die Ladder Logic inkrementieren, oder Sie können 20 einzelne XMIT-Blöcke programmieren und diese anschließend einzeln über die Ladder Logic aktivieren.
Siehe Tabelle zu den Quell- und Zieldatenbereichen für Funktionscodes (21, 21).

Funktionscode	Master-SPS-Datenbereich	Slave-SPS-Datenbereich
20 (Allgemeine Referenz lesen 6x)	4x (Empfänger)	6x (Quelle)
21 (Allgemeine Referenz schreiben 6x)	4x (Quelle)	6x (Empfänger)

XMIT-Kommunikationsblock-Register 4x + 10 bis 4x + 15

(4x + 10) Nachrichten-länge	Sie geben die Länge der aktuellen Meldung ein. Wenn XMIT-Modbus-Meldungen für die Funktionscodes 01, 02, 03, 04, 05, 06, 08, 15 und 16 sendet, wird die Länge der Meldung automatisch auf fünf gesetzt. Wenn XMIT einen abgeschlossenen ASCII-Eingang empfängt, muss die Nachrichtenlänge auf fünf gesetzt werden. Ansonsten tritt ein Fehler auf. Wenn XMIT Modbus-Meldungen für die Funktionscodes 20 und 21 sendet, wird die Länge der Meldung automatisch auf sechs gesetzt. Wenn XMIT ASCII-Meldungen sendet, kann die Länge 1 bis 1024 ASCII-Zeichen pro Meldung betragen.
(4x + 11) Antwort-Timeout (ms)	Hier geben Sie in Millisekunden (ms) einen Wert für die Zeit ein, um festzulegen, wie lange XMIT auf eine gültige Antwortnachricht von einem Slave-Gerät (SPS, Modem usw.) wartet. Außerdem wird die Zeit auf ASCII-Übertragungen und Vorgänge der Datenflusssteuerung angewandt. Wird die Antwortnachricht in der festgelegten Zeit nicht vollständig gebildet, gibt XMIT einen Fehler aus. Der gültige Bereich ist 0 bis 65535 ms. Der Timeout wird nach dem Senden des letzten Zeichens der Nachricht eingeleitet.
(4x + 12) Wiederholungs-grenze	Hier geben Sie die Anzahl der Wiederholungen ein, um festzulegen, wie oft XMIT eine Nachricht sendet, um von einem Slave-Gerät (SPS, Modem usw.) eine gültige Antwort zu erhalten. Wird die Antwortnachricht in der festgelegten Zeit nicht vollständig gebildet, gibt XMIT einen Fehler und einen Fehlercode aus. Der gültige Bereich ist 0 bis 65535 Wiederholungen. Dieses Feld wird zusammen mit dem Antwort-Timeout (4x + 11) verwendet.
(4x + 13) Beginn der Übertragungs-verzögerung (ms)	Hier geben Sie in Millisekunden (ms) den Zeitwert ein, wenn die RTS/CTS-Steuerung aktiviert ist, um festzulegen, wie lange XMIT nach Empfang von CTS wartet, bis es eine Nachricht von der SPS-Schnittstelle überträgt. Sie können dieses Register auch dann verwenden, wenn RTS/CTS NICHT verwendet wird. In dieser Situation bestimmt der eingegebene Zeitwert, wie lange XMIT wartet, bevor es von der SPS-Schnittstelle Nr. 1 eine Nachricht überträgt. Sie können dieses auch als "Vor-Nachricht-Verzögerungszeitgeber" verwenden. Der gültige Bereich ist 0 bis 65535 ms.

(4x + 14) Ende der Übertragungs-verzögerung (ms)	Hier geben Sie in Millisekunden (ms) den Zeitwert ein, wann die RTS/CTS-Steuerung aktiviert ist, um festzulegen, wie lange XMIT RTS belegt hält, bis es eine Nachricht von der SPS-Schnittstelle Nr. 1 überträgt. Wenn die Zeit abgelaufen ist, hält XMIT RTS nicht mehr belegt. Sie können dieses Register auch dann verwenden, wenn RTS/CTS NICHT verwendet wird. In dieser Situation bestimmt der eingegebene Zeitwert, wie lange XMIT wartet, nachdem es von der SPS-Schnittstelle Nr. 1 eine Nachricht überträgt. Sie können dieses auch als "Nach-Nachricht-Verzögerungszeitgeber" verwenden. Der gültige Bereich ist 0 bis 65535 ms.
(4x + 15) Aktuelle Wiederholung – Nur lesen	Der hier angezeigte Wert gibt die aktuelle Anzahl der vom XMIT-Block durchgeführten Wiederholungen an. Dieses Register ist schreibgeschützt.

4.3 Beschreibung und Verwendung des XMIT-Schnittstellenstatusblocks

Auf einen Blick

Zweck

In diesem Abschnitt werden sowohl die Funktionen als auch die Register des Schnittstellenstatusblocks beschrieben.

Inhalt dieses Abschnitts

Dieser Abschnitt enthält die folgenden Themen:

Thema	Seite
XMIT-Schnittstellenstatusblock und SPS-Kompatibilität	105
Darstellung des XMIT-Funktionsblocks für den Schnittstellenstatus und Inhalt des oberen Eintrags	107
Tabelle der XMIT-Schnittstellenstatusanzeigen	109

XMIT-Schnittstellenstatusblock und SPS-Kompatibilität

SPS-Kompatibilität

Der XMIT-Schnittstellenstatusblock ist mit folgenden SPS von Schneider Electric kompatibel.

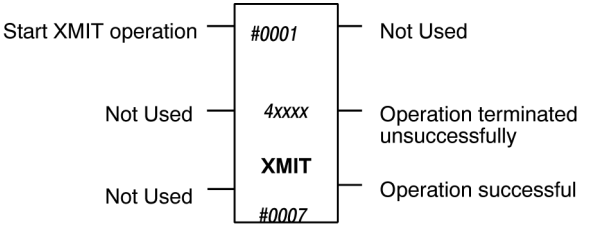
Produktfamilie	Modellnummern
Quantum	140CPU11302
	140CPU21304
	140CPU11303 (mit Hauptsteuerprogramm 2.12 oder höher)
	140CPU42402 (mit Hauptsteuerprogramm 2.10 oder höher)
	140CPU43412
	140CPU53414 (mit Hauptsteuerprogramm 1.02 oder höher)
	140CPU43412A
	140CPU53414A
Compact	PC E984 241
	PC E984 245
	PC E984 251
	PC E984 255 (mit Hauptsteuerprogramm 1.02 oder höher)
	984-E285
	984-E265
	984-E275
	984-285
Micro	110CPU61204 (mit Hauptsteuerprogramm 1.00 oder höher)

Produktfamilie	Modellnummern
Momentum	171CCS70000
	171CCS70010
	171CCS76000
	171CCS78000
	171CCC76010
	171CCC78010 (mit Hauptsteuerprogramm 2.00 oder höher)
	171CCC98020
	171CCC96020
	171CCC98030
	171CCC96030
	171CCC98091
	171CCC96091
Momentum-Steuerungen unterstützen nur ein Stoppbit.	

Darstellung des XMIT-Funktionsblocks für den Schnittstellenstatus und Inhalt des oberen Eintrags

Darstellung

Die Höhe des Blocks beträgt drei Einträge. Struktur des XMIT-Schnittstellenstatusblocks



Eingänge

Der XMIT-Schnittstellenstatusblock verfügt über einen möglichen Steuerungseingang. Der Eingang für den oberen Eintrag startet einen XMIT-Vorgang und sollte so lange auf ON gesetzt bleiben, bis der Vorgang erfolgreich abgeschlossen wurde oder ein Fehler aufgetreten ist. Der Eingang des mittleren Eintrags wird im XMIT-Schnittstellenstatusblock nicht verwendet.

Hinweis: Um einen XMIT-Fehler zurückzusetzen und das Fehlerregister zu löschen, muss der obere Eingang für mindestens einen SPS-Programmzyklus auf OFF geschaltet werden.

Ausgänge

Der XMIT-Schnittstellenstatusblock kann zwei mögliche Ausgänge erstellen. Der Ausgang des oberen Eintrags wird im XMIT-Schnittstellenstatusblock nicht verwendet. Der Ausgang vom mittleren Eintrag wird auf ON geschaltet, wenn XMIT einen Fehler erkannt hat oder ein Abbruchbefehl gegeben wurde. Der Ausgang vom unteren Eintrag wechselt in den Zustand ON, wenn ein XMIT-Vorgang erfolgreich abgeschlossen wurde.

Inhalt des oberen Eintrags

Der obere Eintrag muss eine der folgenden Konstanten enthalten:

- (#0001) zur Auswahl von SPS-Schnittstelle Nr. 1
- (#0002) zur Auswahl von SPS-Schnittstelle Nr. 2

WICHTIG:

- LOADABLE AKZEPTIERT 4x-Register im oberen Eintrag.
- INTEGRATION AKZEPTIERT KEINE 4x-Register im oberen Eintrag.

Inhalt des mittleren Eintrags

Das im mittleren Eintrag eingegebene 4x-Register ist das erste in einer Gruppe von sieben (7) zusammenhängenden Ausgangsregistern, die den Schnittstellenstatusanzeigeblock, wie unten dargestellt, beinhalten:
Beschreibung der Tabelle für die XMIT-Schnittstellenstatussteuerung

Register	Beschreibung	Keine gültigen Einträge
4x	XMIT-Versionsnummer	Nur lesen
4x + 1	Fehlerstatus	Nur lesen
4x + 2	Slave-Anmeldestatus/Aktiver Schnittstellenzustand des Slave	Nur lesen
4x + 3	Anzahl der Slave-Transaktionen	Nur lesen
4x + 4	Port-Status	Nur lesen
4x + 5	Statusbit des Eingangs-FIFO	Nur lesen
4x + 6	Maximale Länge des Eingangs-FIFO	Nur lesen

Hinweis: ADRESSE NICHT ÄNDERN
Ändern Sie KEINESFALLS die Adresse im mittleren Eintrag des XMIT-Blocks, und löschen Sie sie nicht aus dem Programm, wärend es aktiv ist. Anderenfalls wird die Schnittstelle blockiert, und die Kommunikation wird unterbrochen.

Inhalt des unteren Eintrags

Der untere Eintrag muss eine Konstante enthalten, die (#0007) entspricht. Dies ist die Anzahl der Register, die durch die XMIT-Schnittstellenstatusanweisung verwendet wird.

Tabelle der XMIT-Schnittstellenstatusanzeigen

(4x) XMIT-Versionsnummer – Schreibgeschützt

Zeigt die aktuelle Versionsnummer des XMIT-Blocks an. Diese Nummer wird automatisch vom Block geladen, und der Block überschreibt alle anderen in dieses Register eingegebenen Nummern.

(4x + 1) Fehlerzustand des Schnittstellenstatus – Schreibgeschützt

Dieses Feld zeigt einen vom XMIT-Schnittstellenstatusblock generierten Fehlercode an. Die folgende Tabelle enthält eine vollständige Auflistung. Siehe Fehlerstatustabelle (4x + 1).

Fehlercode	Fehlerbeschreibung
118	XMIT konnte nicht auf die SPS-Kommunikationsschnittstelle Nr. 1 oder Nr. 2 zugreifen
122	Der obere Eintrag von XMIT ist nicht gleich null, eins oder zwei
123	Der untere Eintrag von XMIT ist nicht gleich sieben, acht oder sechzehn

(4x + 2) Slave-Anmeldestatus/ Aktiver Status des Schnittstellen-Slave – Schreibgeschützt

Das Register (4x + 2) des XMIT-Schnittstellenstatusblocks erstellt sowohl den Slave-Anmeldestatus als auch den aktiven Status des Schnittstellen-Slave und zeigt diese an. Die Ladder Logic kann diese Informationen verwenden, um Datenkollisionen in einem Modbus-Netzwerk mit mehreren Mastern zu reduzieren oder zu vermeiden.
Statusberichte des Registers (4x + 2)

(4x + 2 höherwertiges Byte) Slave-Anmeldestatus	(4x + 2 niederwertiges Byte) Aktiver Schnittstellenzustand
Ja = Wenn ein Programmiergerät derzeit an der Slave-Schnittstelle dieser SPS angemeldet ist	Ja = Wenn eine überwachte Schnittstelle zu einer SPS gehört und derzeit einen Modbus-Befehl empfängt ODER eine Modbus-Antwort überträgt
Nein = Wenn ein Programmiergerät derzeit NICHT an der Slave-Schnittstelle dieser SPS angemeldet ist (HINWEIS: Ein Modbus-Master kann Befehle senden, kann jedoch nicht angemeldet werden)	Nein = Wenn eine überwachte Schnittstelle NICHT zu einer SPS gehört und derzeit einen Modbus-Befehl empfängt ODER eine Modbus-Antwort überträgt

(4x + 3) Slave-Transaktionszähler – Schreibgeschützt

Dieses Register zeigt die Anzahl der Slave-Transaktionen an, die vom XMIT-Schnittstellenstatusblock erstellt werden. Der Zähler wird immer dann erhöht, wenn die Modbus-Slave-Schnittstelle der SPS einen weiteren Befehl vom Modbus-Master erhält. Die Ladder Logic kann diese Informationen verwenden, um Datenkollisionen in einem Modbus-Netzwerk mit mehreren Mastern zu reduzieren oder zu vermeiden.

(4x + 4) Schnittstellenstatus – Schreibgeschützt

Dieses Register zeigt das privilegierte Schreibrecht der Schnittstelle und deren Zustand an. Es wird durch den XMIT-Schnittstellenstatusblock generiert. Register (4x + 4) – Tabelle mit den Schnittstellenstatusoptionen.

Privilegiertes Schreibrecht für Schnittstelle	Aktiver Status	Wert
SPS	SPS-Modbus-Slave	0
XMIT	Tonwahlmodem	1
XMIT	Modem auflegen	2
XMIT	Modbus-Meldungen	3
XMIT	Einfacher ASCII-Ausgang	4
XMIT	Impulswahlmodem	5
XMIT	Modem initialisieren	6
XMIT	Einfacher ASCII-Eingang	7
XMIT	Abgeschlossener ASCII-Eingang	8
XMIT	ASCII-Eingangs-FIFO ist auf ON geschaltet, es ist jedoch KEINE XMIT-Funktion aktiv	9

(4x + 5) Statusbits des Eingangs-FIFO – Schreib-geschützt

Das Register (4x + 5) zeigt den Status von sieben Elementen in Bezug auf das Eingangs-FIFO an. Es wird durch den XMIT-Schnittstellenstatusblock generiert. Siehe Eingangs-FIFO-Statusbits (4x + 5) und die zugehörige Definitionstabelle.

Bit-Nr.	Definition	Ja/1=	Nein/0=
1 ... 3	Nicht benutzt		
4	Privilegiertes Schreibrecht für die Schnittstelle liegt bei...	XMIT	SPS
5 ... 7	Nicht benutzt		
8	ASCII-Ausgangsübertragung...	Durch Empfangsgerät gesperrt	Durch Empfangsgerät freigegeben
9	ASCII-Eingang empfangen...	Neues Zeichen	KEIN neues Zeichen
10	ASCII-Eingangs-FIFO ist...	Leer	Nicht leer
11	ASCII-Eingangs-FIFO ist...	Im Überlaufzustand (Fehler)	Nicht im Überlaufzustand (Fehler)
12	ASCII-Eingangs-FIFO ist...	Ein	Aus
13 ... 15	Nicht benutzt		
16	ASCII-Eingangsempfang...	Sendendes Gerät durch XMIT gesperrt	Sendendes Gerät durch XMIT freigegeben

(4x + 6) Länge der FIFO-Eingangsdaten – Schreib-geschützt

Dieses Register zeigt die aktuelle Anzahl der Zeichen an, die im ASCII-Eingangs-FIFO vorhanden sind. Das Register kann je nach Status des Eingangs-FIFO und je nachdem, ob die Datenlänge leer ist oder sich im Überlaufzustand befindet, andere Werte enthalten. Es wird durch den XMIT-Schnittstellenstatusblock generiert. Siehe Tabelle mit anderen möglichen Werten für (4x + 6).

Zustand des Eingangs-FIFO	Länge
= AUS	= 0
= EIN & leer	= 0
= EIN & Überlauf	= 512

4.4 Beschreibung des XMIT-Konvertierungsblocks

Auf einen Blick

Zweck

In diesem Abschnitt werden die Kompatibilität des Konvertierungsblocks mit Produkten von Schneider Electric sowie die Darstellung des Blocks und der Inhalt der Einträge beschrieben.

Inhalt dieses Abschnitts

Dieser Abschnitt enthält die folgenden Themen:

Thema	Seite
XMIT-Konvertierungsblock und SPS-Kompatibilität	113
Struktur und Inhalt des XMIT-Konvertierungsblocks	115

XMIT-Konvertierungsblock und SPS-Kompatibilität

SPS-Kompatibilität

Der XMIT-Schnittstellenstatusblock ist mit folgenden SPS von Schneider Electric kompatibel.

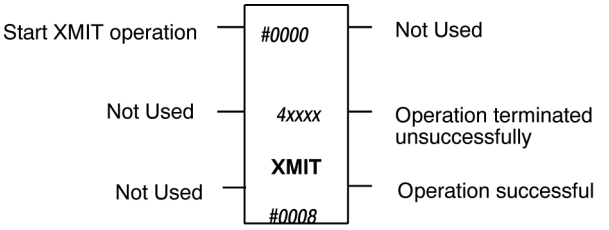
Produktfamilie	Modellnummern
Quantum	140CPU11302
	140CPU21304
	140CPU11303 (mit Hauptsteuerprogramm 2.12 oder höher)
	140CPU42402 (mit Hauptsteuerprogramm 2.10 oder höher)
	140CPU43412
	140CPU53414 (mit Hauptsteuerprogramm 1.02 oder höher)
	140CPU43412A
	140CPU53414A
Compact	PC E984 241
	PC E984 245
	PC E984 251
	PC E984 255 (mit Hauptsteuerprogramm 1.02 oder höher)
	984-E285
	984-E265
	984-E275
	984-285
Micro	110CPU61204 (mit Hauptsteuerprogramm 1.00 oder höher)

Produktfamilie	Modellnummern
Momentum	171CCS70000
	171CCS70010
	171CCS76000
	171CCS78000
	171CCC76010
	171CCC78010 (mit Hauptsteuerprogramm 2.00 oder höher)
	171CCC98020
	171CCC96020
	171CCC98030
	171CCC96030
	171CCC98091
	171CCC96091
Momentum-Steuerungen unterstützen nur ein Stoppbit.	

Struktur und Inhalt des XMIT-Konvertierungsblocks

Darstellung

Struktur des XMIT-Konvertierungsblocks. Die Höhe des XMIT-Konvertierungsblocks beträgt drei Einträge.



Eingänge

XMIT verfügt über einen möglichen Steuerungseingang. Der Eingang für den oberen Eintrag startet einen XMIT-Vorgang und sollte so lange im Zustand ON bleiben, bis der Vorgang erfolgreich abgeschlossen wurde oder ein Fehler aufgetreten ist. Der Eingang für den mittleren Eintrag wird für den XMIT-Konvertierungsblock nicht verwendet.

Hinweis: Um einen XMIT-Fehler zurückzusetzen und das Fehlerregister zu löschen, muss der obere Eingang für mindestens einen SPS-Programmzyklus in den Zustand OFF geschaltet werden.

Ausgänge

XMIT kann zwei mögliche Ausgänge erstellen. Der Ausgang des oberen Eintrags wird im XMIT-Konvertierungsblock nicht verwendet. Der Ausgang vom mittleren Eintrag wird in den Zustand ON versetzt, wenn XMIT einen Fehler erkannt hat oder ein Abbruchbefehl gegeben wurde. Der Ausgang vom unteren Eintrag wechselt in den Zustand ON, wenn ein XMIT-Vorgang erfolgreich abgeschlossen wurde.

Inhalt des oberen Eintrags

Der obere Eintrag muss eine Konstante enthalten (#0000), da Konvertierungen nicht mit der SPS-Schnittstelle interagieren.
WICHTIG:

- LOADABLE AKZEPTIERT 4x-Register im oberen Eintrag.
- INTEGRATION AKZEPTIERT KEINE 4x-Register im oberen Eintrag.

Inhalt des mittleren Eintrags

Das im mittleren Eintrag eingegebene 4x-Register ist das erste in einer Gruppe von acht (8) zusammenhängenden Ausgangsregistern, die den Steuerungsblock, wie unten dargestellt, beinhalten:
XMIT-Konvertierungssteuerungstabelle

Register	Beschreibung	Gültige Einträge
4x	XMIT-Versionsnummer	Nur lesen
4x + 1	Fehlerstatus	Nur lesen
4x + 2	Verfügbar für Benutzer	0 (Kann als Zeiger für Anweisungen, wie beispielsweise TBLK, verwendet werden)
4x + 3	Steuerungsbit für die Datenkonvertierung	Siehe Bit-Definitionstabelle für 4x + 3.
4x + 4	Datenkonvertierungs-Opcode	Siehe Bit-Definitionstabelle für 4x + 4.
4x + 5	Quellregister	4x-Register (Lesevorgang beim höherwertigen oder niederwertigen Byte beginnen)
4x + 6	Zielregister	4x-Register (Lesevorgang beim höherwertigen oder niederwertigen Byte beginnen)
4x + 7	Anzahl der ASCII-Zeichenfolgen	Legt den Suchbereich fest

Hinweis: ADRESSE NICHT ÄNDERN
Ändern Sie KEINESFALLS die Adresse im mittleren Eintrag des XMIT-Blocks, und löschen Sie sie nicht aus dem Programm, wärend es aktiv ist. Anderenfalls wird die Schnittstelle blockiert, und die Kommunikation wird unterbrochen.

Inhalt des unteren Eintrags

Der untere Eintrag muss eine Konstante enthalten, die (#0008) entspricht. Dies ist die Anzahl der Register, die durch die XMIT-Konvertierungsanweisung verwendet wird.

4.5 Verwendung des XMIT-Konvertierungsblocks

Auf einen Blick

Zweck

In diesem Abschnitt werden die Register 4x bis 4x + 7 des Konvertierungsblocks beschrieben.

Inhalt dieses Abschnitts

Dieser Abschnitt enthält die folgenden Themen:

Thema	Seite
XMIT-Konvertierungsblock-Register 4x bis 4x + 2	118
XMIT-Konvertierungsblock, Register 4x + 3	119
XMIT-Konvertierungsblock-Register 4x + 4 bis 4x + 7	121

XMIT-Konvertierungsblock-Register 4x bis 4x + 2

(4x) XMIT-Versionsnummer – Schreibgeschützt

Das Register (4x), XMIT-Versionsnummer, zeigt die aktuelle Versionsnummer des XMIT-Blocks an. Der Block lädt automatisch die aktuelle Versionsnummer in das Register. Daher überschreibt der Block alle vorigen Versionsnummern, die in dieses Register eingegeben wurden.

(4x + 1) Konvertierungsfehlerstatus – Schreibgeschützt

Dieses Feld zeigt einen vom XMIT-Konvertierungsblock generierten Fehlercode an. Die folgende Tabelle enthält eine vollständige Auflistung.
Fehlerstatus (4x + 1) – Schreibgeschützt – Tabelle.

Fehlercode	Fehlerbeschreibung
122	Der obere Eintrag von XMIT ist nicht gleich null, eins oder zwei
123	Der untere Eintrag von XMIT ist nicht gleich sieben, acht oder sechzehn
131	Ungültige Zeichenzählung
135	Ungültiger Zielregisterblock
136	Ungültiger Quellregisterblock
137	Keine ASCII-Nummer vorhanden
138	Mehrere Vorzeichen vorhanden
139	Numerischer Überlauf erkannt
140	Fehlerhafte Zeichenfolge
141	Zeichenfolge nicht gefunden
142	Ungültige Fehlerprüfung erkannt
143	Ungültiger Konvertierungs-Opcode

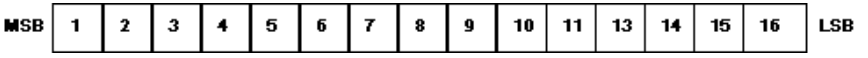
(4x + 2) Verfügbar für Benutzer

Der XMIT-Konvertierungsblock verwendet dieses Register nicht. Es wird jedoch unter Umständen als Zeiger in der Ladder Logic verwendet. Der XMIT-Block lässt sich effizient verwenden, indem ein Zeigerwert einer TBLK-Anweisung in diesem Register platziert wird.

XMIT-Konvertierungsblock, Register 4x + 3

(4x + 3) Steuerungsbit für die Datenkonvertierung

Dieses 16-Bit-Wort bezieht sich auf das Datenkonvertierungswort (4x + 4). Diese Bit bieten zusätzliche Steuerungsoptionen, je nachdem, welche der elf Konvertierungsmöglichkeiten Sie auswählen.
Bit-Verteilung



(4x + 3) Definitionstabelle der Steuerungsbit für die Datenkonvertierung

Bit-Nr.	Definition	1=	0=
1	Nicht benutzt		
2	CRC, 16 Seeds	0x0000	0xFFFF
3	Fehlerprüfungstyp	LRC 8	CRC 16
4	Fehlerprüfung	Validieren	Anhängen
5 & 6	Nicht benutzt		
7	Konvertierungsfall	Oben nach unten	Unten nach oben
8	Groß- und Kleinschreibung	Nein	Ja
9	Format führend	Nullen	Leerzeichen
10	Ausgangsformat	Fest	Variable
11	Konvertierungstyp	Ohne Vorzeichen	Mit Vorzeichen
12	Konvertierungswort	32 Bit	16 Bit
13	Quellenzeiger automatisch erweitern (zeigt auf das nächste Zeichen nach dem zuletzt gelöschten Zeichen)	Ja	Nein
14	Zielzeiger automatisch erweitern (zeigt auf das nächste Zeichen nach dem zuletzt gelöschten Zeichen)	Ja	Nein
15	Lesen von ASCII in Quelle beginnend mit ...	Niederwertiges Byte:	Höherwertiges Byte (Normal)
16	Speichern von ASCII in Ziel beginnend mit ...	Niederwertiges Byte:	Höherwertiges Byte (Normal)

Bestimmte Bit beziehen sich auf bestimmte Konvertierungen. Die nicht erwähnten Bit werden durch die ausgewählte Konvertierung nicht validiert oder geändert, und sie erfüllen keine Funktion in Bezug auf die ausgewählte Konvertierung. Aus diesem Grund werden sie einfach ignoriert.

Konvertierung in Tabelle mit relevanter Bit-Beziehung.

Konvertierungstyp (Opcode)	**Relevanter Bit-Status (Software legt Bit-Status fest)**
Ungültiger Opcode (wird angezeigt, wenn ein ungültiger Opcode erkannt wird)	
ASCII dezimal in Ganzzahl (1)	16=0,11,12,13,15 (7=1, 8=0)
ASCII hexadezimal in Ganzzahl (2)	16=0,11,12,13,15 (7=1, 8=0)
ASCII hexadezimal in Ganzzahlmatrix (3)	13,15,16 (keine)
Ganzzahl in ASCII dezimal (4)	15,9,10,11,12,14,16 (keine)
Ganzzahl in ASCII dezimal (5)	15,9,10,11,12,14,16 (keine)
Ganzzahlmatrix in ASCII hexadezimal (6)	14, 15, 16, (8 = ja)
Quellbyte in Ziel umschalten (7)	14, 15, 16, (8 = ja)
Quellblock in Ziel kopieren (8)	7, 8, 14, 15, 16 (keine)
Quell- & Zielblock vergleichen (9)	7, 8, 13, 15, 16 (keine)
Quellblock nach in Ziel definierter Zeichenfolge durchsuchen (10)	7, 8, 13, 15, 16 (keine)
Validieren oder Fehlerprüfung in Quellblock anhängen (11)	2, 3, 4, 13, 15 (8 = ja, 14 = ja, 16 = 1/0)

XMIT-Konvertierungsblock-Register 4x + 4 bis 4x + 7

(4x + 4) Daten-konvertierungs-Opcodes

Wählen Sie in der Liste der elf Optionen in der Tabelle den Konvertierungstyp aus, den Sie ausführen möchten. Nach der Auswahl des Konvertierungstyps finden Sie in den Steuerungsbit für die Datenkonvertierung (4x + 3) weitere Steuerungsoptionen, die sich speziell auf den ausgewählten Konvertierungstyp beziehen.
(4x + 4) Definitionstabelle der Opcodes für die Datenkonvertierung

Datentyp (4x-Block)	Aktion	Datentyp (4x-Block)
Unzulässiger Opcode	wird angezeigt, wenn ein unzulässiger Opcode erkannt wird	Nicht zutreffend
ASCII-Zeichenfolge im Dezimalformat empfangen (1)	Konvertiert in	Binäre 16-Bit- oder 32-Bit-Ganzzahl mit oder ohne Vorzeichen
ASCII-Zeichenfolge im Hexadezimalformat empfangen (2)		Binäre 16-Bit- oder 32-Bit-Ganzzahl ohne Vorzeichen
ASCII-Zeichenfolge im Hexadezimalformat empfangen (3)		Binäre 16-Bit-Ganzzahlmatrix ohne Vorzeichen
16-Bit- oder 32-Bit-Ganzzahl mit oder ohne Vorzeichen (4)		ASCII-Zeichenfolge im Dezimalformat zur Übertragung
Binäre16-Bit- oder 32-Bit-Ganzzahl ohne Vorzeichen (5)		ASCII-Zeichenfolge im Hexadezimalformat zur Übertragung
16-Bit-Ganzzahlmatrix ohne Vorzeichen		ASCII-Zeichenfolge im Hexadezimalformat zur Übertragung
Hohes und niederwertiges Byte des gespeicherten ASCII-Quellregisterblocks (7)	Ausgetauscht gegen	ASCII-Zielregisterblock
ASCII-Zeichenfolge aus dem Quellregisterblock (8)	Kopiert in	ASCII-Zielregisterblock mit oder ohne Konvertierung der Groß-/Kleinschreibung
ASCII-Quellregisterblock (9)	Verglichen mit	Im Zielregisterblock definierte ASCII-Zeichenfolge mit oder ohne Berücksichtigung der Groß- und Kleinschreibung
ASCII-Quellregisterblock (10)	Suchen nach	Im Zielblock definierte ASCII-Zeichenfolge mit oder ohne Berücksichtigung der Groß- und Kleinschreibung
Fehlerprüfung für 8bit-LRC oder 16bit-CRC (11)	Validiert oder Angehängt an	ASCII-Zeichenfolge im Quellregisterblock

Hinweis: Konvertierungen von binär in BCD und von BCD in binär können unter Verwendung eines oder mehrerer XMIT-Konvertierungsblocks vorgenommen werden. Details siehe *XMIT Binär/BCD-Konvertierungstypen, S. 132*.

(4x + 5) Quellregister

Geben Sie das gewünschte 4x-Register ein. Dies ist das erste Register des Quellblocks, das gelesen wird. Stellen Sie sicher, dass Sie auswählen, wo der Lesevorgang beginnen soll (höherwertiges oder niederwertiges Byte). Die Auswahl neben diesem Register im DX-Zoomfenster entspricht Bit 15 in (4x +3).

(4x + 6) Zielregister

Geben Sie das gewünschte 4x-Register ein. Dies ist das erste Register des Zielblocks, das gespeichert wird. Stellen Sie sicher, dass Sie auswählen, wo der Speichervorgang beginnen soll (höherwertiges oder niederwertiges Byte). Die Auswahl neben diesem Register im DX-Zoom entspricht Bit 16 in (4x +3).

(4x + 7) Anzahl der ASCII-Zeichenfolgen

Geben Sie den Suchbereich ein. Im Register ist der Suchbereich festgelegt. Wenn die automatische Erweiterung der Quelle (Bit 13) oder die automatische Erweiterung des Ziels (Bit 14) auf ON gesetzt ist und kein ASCII-Zeichen erkannt wird, passt der Block die Zeichenzählung automatisch an.

4.6 Beispiele für den XMIT-Konvertierungsblock-Opcode

Auf einen Blick

Zweck

In diesem Abschnitt werden elf Opcode-Beispiele dargestellt.

Inhalt dieses Abschnitts

Dieser Abschnitt enthält die folgenden Themen:

Thema	Seite
XMIT-Konvertierungs-Opcode – Beispiele 1 bis 3	124
XMIT-Konvertierungsblock-Opcode – Beispiele 4 bis 6	127
XMIT-Konvertierungsblock-Opcode – Beispiele 7 bis 11	129
XMIT Binär/BCD-Konvertierungstypen	132

XMIT-Konvertierungs-Opcode – Beispiele 1 bis 3

Konvertierungsbeispiele bezogen auf ASCII-Eingang

Die Opcodes 1 bis 3 konvertieren ASCII-Eingangsdaten in Binärdaten. Die ASCII-Eingangsdaten werden über die SPS-Schnittstelle und den XMIT-Kommunikationsblock empfangen. Anschließend werden die ASCII-Daten in Binärdaten konvertiert. Dann sind die konvertierten Binärdaten bereit für die Verwendung durch die SPS, die Ihren Anwendungsanforderungen entspricht.
Diese Opcodes analysieren ASCII-Zeichenkettendaten veränderlicher Länge, beginnend im höherwertigen oder niederwertigen Byte des Quellregisters, je nachdem, was im Register für die Datenkonvertierungssteuerung (4x +3, Bit 15) ausgewählt ist. Das Zeichenzählungsregister für ASCII-Zeichenfolgen (4x + 7) definiert die maximale Zeichenzahl, die in der Quellzeichenfolge analysiert werden kann. Sie muss anfänglich einen Wert zwischen 1 und 1024 enthalten. Durch das Register zur Steuerung der Datenkonvertierung (4x + 3) werden außerdem die Konvertierungslänge von 16 Bit oder 32 Bit (4x + 3, Bit 12) und die Konvertierung mit oder ohne Vorzeichen (4x +3, Bit 13) ausgewählt.
Für die Opcodes 1 bis 3 wird die anfängliche Zeichenzählung der ASCII-Zeichenfolge (4x + 7) durch die Gesamtanzahl der analysierten Zeichen in der ASCII-Quellzeichenfolge reduziert, und der Zeiger für die Quellzeichenfolge wird auf ein Zeichen nach dem letzten während der Konvertierung analysierten Zeichen erweitert.

Hinweis: Falls keine Hexadezimal- oder Dezimalzahlen vorhanden sind oder wenn der Zielregisterblock (4x + 6) mehr als 512 Register umfasst oder das Ende der RAM-Konfiguration für den SPS-Status überschreitet, tritt ein Fehler auf.

Tabelle der Konvertierungsbeispiele bezogen auf ASCII-Eingang

Opcode	Aktionen	Daten (von Ihnen eingegeben)
1	Quellblock, beginnend beim höherwertigen Byte 400201 =	"-001234567crlf"
	Anfängliche Zeichenzählung =	0x000C
	Auswahl für Konvertierungssteuerung =	32-Bit-Konvertierung mit Vorzeichen
	32 Bit, mit denen Zielregisterpaar geladen ist =	0xFFED2979
	Quellblock, erweitert auf das höherwertige Byte 400206, zielt jetzt auf =	"crlf"
	Zeichenzählung der ASCII-Zeichenfolge wird reduziert auf =	0x0002

Opcode	Aktionen	Daten (von Ihnen eingegeben)
2	Quellblock, beginnend beim höherwertigen Byte 400201 =	"+F301C23 cat"
	Anfängliche Zeichenzählung =	0x000C
	Auswahl für Konvertierungssteuerung =	32-Bit-Konvertierung ohne Vorzeichen
	32 Bit, mit denen Zielregisterpaar geladen ist =	0x0FE01C23
	Quellblock, erweitert auf das höherwertige Byte 400205, zielt jetzt auf =	"cat"
	Zeichenzählung der ASCII-Zeichenfolge wird reduziert auf =	0x0004
3	Quellblock, beginnend beim niederwertigen Byte 400301 =	"124ABC0AFcrlf"
	Anfängliche Zeichenzählung =	0x000B
	32 Bit, mit denen Zielregisterpaar geladen ist =	0x0FE01C23
	Quellblock, erweitert auf das höherwertige Byte 400306, zielt jetzt auf =	"crlf"
	Zeichenzählung der ASCII-Zeichenfolge wird reduziert auf =	0x0002

Beschreibung des Beispiels für Opcode 1

Opcode 1 überspringt anfängliche Leerzeichen und sucht nach dem optionalen Zeichen "+" oder "-" und nach mindestens einer Dezimalzahl zwischen "0" und "9" und einem Ende, das weder ein Leerzeichen noch eine Dezimalzahl ist. Anschließend wird der entsprechende binäre Wert der Zeichenfolge zur 16-Bit-Konvertierung in das Zielregister oder zur 32-Bit-Konvertierung in das Zielregisterpaar geschrieben. Im 32-Bit-Zielregisterpaar ist das niederstwertige Wort unter der niedrigeren Registernummer und das höchstwertige Wort unter der höheren Registernummer gespeichert. Wenn keine Dezimalzahl vorhanden ist oder so viele Ziffern vorhanden sind, dass die konvertierte binäre Entsprechung zu groß für den angeforderten Speichertyp ist, tritt ein Fehler auf.

Beschreibung des Beispiels für Opcode 2

Opcode 2 überspringt anfängliche Leerzeichen und sucht nach dem optionalen Zeichen "+" oder "-" und nach mindestens einer Hexadezimalzahl zwischen "0" und "9" oder "A" und "F" bzw. "a" und "f" und einem Ende, das weder ein Leerzeichen noch eine Hexadezimalzahl ist. Anschließend wird der entsprechende binäre Wert der Zeichenfolge in das 16-Bit- oder in das 32-Bit-Ziel geschrieben.

Beschreibung des Beispiels für Opcode 3

Opcode 3 konvertiert ASCII-Hexadezimalzeichen in eine Reihe binärer 16-Bit-Entsprechungen, wobei jeweils 4 ASCII-Zeichen in ein 16-Bit-Speicherwort gepackt werden.

Konvertierungsbeispiel bezogen auf ASCII-Ausgang

Die Opcodes 4 bis 6 konvertieren SPS-Binärdaten in ASCII-Daten. Sobald die SPS-Binärdaten in ASCII-Daten konvertiert wurden, werden sie über die SPS-Schnittstelle und den XMIT-Kommunikationsblock übertragen. Dann sind die konvertierten ASCII-Daten bereit für die Verwendung durch das Feldgerät, das Ihren Anwendungsanforderungen entspricht.

Hinweis: In den Opcodes 4 bis 6 tritt ein Fehler auf, wenn der Zielregisterblock mehr als 512 Register umfasst oder das Ende der RAM-Konfiguration für den SPS-Status überschreitet.

Tabelle der Konvertierungsbeispiele bezogen auf ASCII-Ausgang

Opcode	Aktionen	Daten (von Ihnen eingegeben)
4	Quelle enthält =	0x9CDE
	Zielblock beim höherwertigen Byte 400101	
	Anfängliche Zeichenzählung =	0x000C
	Die Konvertierungssteuerung wählt das feste 16-Bit-Ausgangsformat mit Vorzeichen unter Verwendung führender Nullen aus	
	Zielblock ist geladen mit =	"000000040158"
	Endgültige Zeichenzählung =	0x0000
	Zielblock erweitert auf höherwertiges Byte 400107	
5	Quelle enthält =	0x03FE1234
	Zielblock beim niederwertigen Byte 400001	
	Anfängliche Zeichenzählung =	0x0010
	Die Konvertierungssteuerung wählt das veränderliche 32-Bit-Ausgangsformat ohne Vorzeichen aus	
	Zielblock ist geladen mit =	"3FE1234"
	Endgültige Zeichenzählung =	0x0009
	Zielblock erweitert auf höherwertiges Byte 400005	
6	Quelle enthält =	0x5B3D, 0x467E, 0xD14F, 0x478C
	Zielblock beim niederwertigen Byte 400201	
	Anfängliche Zeichenzählung =	0x0007
	Zielblock ist geladen mit =	"5B3D467"
	Endgültige Zeichenzählung =	0x0000
	Zielblock erweitert auf höherwertiges Byte 400205	

XMIT-Konvertierungsblock-Opcode – Beispiele 4 bis 6

Beschreibung des Beispiels für Opcode 4 und 5

Die Opcodes 4 und 5 erstellen ASCII-Ausgangsdaten veränderlicher Länge, wenn das Steuerungsregister für die Datenkonvertierung (4x +3, Bit 10) auf 0 (variabel) gesetzt ist. Dann wird die Anzahl der ASCII-Ausgangszeichen erstellt und vom anfänglichen Wert des ASCII-Zeichenfolgenzählungsregisters (4x + 7) subtrahiert. Der Zielzeiger (4x + 3, Bit 14) wird erweitert. Wenn das Steuerungsregister für die Datenkonvertierung (4x +3, Bit 10) auf 1 (fest) gesetzt ist, geschieht Folgendes: Basierend auf dem Status des Steuerungsregisters für die Datenkonvertierung (4x +3, Bit 9) werden ausreichend Nullen und Leerzeichen in den Zielregisterblock (4x + 6) vor den Konvertierungsdaten geladen, um zu erzwingen, dass die gesamte Anzahl der Zeichen genau dem angeforderten Betrag entspricht. Die Zeichenzählung der ASCII-Zeichenfolge (4x + 7) wird auf null gesetzt, und der Zielzeiger (4x + 3, Bit 14) wird erweitert. Wenn der Wert der Binärquelle mehr Dezimalzeichen erstellt als in den Zielregisterblock passen, tritt ein Fehler auf.

Beschreibung des Beispiels für Opcode 6

Opcode 6 konvertiert eine Matrix binärer Register des Quellblocks in ASCII-Hexadezimalzeichen, die im Zielblock geladen werden.

Konvertierungsbeispiele bezogen auf die ASCII-Zeichenfolge

Opcodes 7 bis 11 führen fünf unterschiedliche ASCII-Zeichenfolgenvorgänge innerhalb der SPS gemäß den Anforderungen Ihrer Anwendung aus. Wir empfehlen Ihnen, Ihre Quell- und Zielblöcke unter Verwendung unterschiedlicher 4x-Referenzen zu definieren, die sich nicht überschneiden.
Wenn Sie einen Byte-Austausch (Opcode 7) oder eine Kopie einer Zeichenfolge (Opcode 8) mit Groß- und Kleinschreibungskonvertierung verwenden, sind der Quell- und der Zielblock unter Umständen identisch. Wenn Sie einen Byte-Austausch (Opcode 7) oder eine Kopie einer Zeichenfolge (Opcode 8) verwenden, wird der Zielblock geladen, der Zielzeiger (4x + 3, Bit 14) über das letzte geschriebene Zeichen hinaus erweitert, und die Zeichenzählung der ASCII-Zeichenfolge (4x + 7) wird auf null reduziert.
Wenn Sie einen Zeichenfolgenabgleich (Opcode 9) oder eine Zeichenfolgensuche (Opcode 10) verwenden, wird der Quellzeiger (4x + 3, Bit 13) erweitert, und die Zeichenzählung der ASCII-Zeichenfolge (4x + 7) wird reduziert.

Hinweis: Im Allgemeinen müssen die automatische Erweiterung für den Quellzeiger (4x + 3, Bit 13) und die automatische Erweiterung für den Zielzeiger (4x + 3, Bit 14) im Konvertierungssteuerregister (4x + 3) aktiviert sein. Ansonsten behalten diese Zeiger ihre ursprünglichen Werte sowie die ursprüngliche Zeichenzählung (4x + 7) bei.

Tabelle mit Beispielen für die Konvertierung der ASCII-Zeichenfolge.

Opcode	**Aktionen**	**Daten (von Ihnen eingegeben)**
7	Quelle enthält =	"ABCDEF"
	Zielblock beim niederwertigen Byte 400001	
	Anfängliche Zeichenzählung =	0x0006
	Zielblock ist geladen mit =	"BADCFE"
	Abschließende Zeichenzählung wird reduziert auf =	0x0000
	Zielblock erweitert auf niederwertiges Byte 400004	
8a	Quelle enthält =	"ABcdeFGH"
	Zielblock beim niederwertigen Byte 400101	
	Anfängliche Zeichenzählung =	0x0006
	Groß-/Kleinschreibung für Konvertierungssteuerung aktiviert	
	Zielblock ist geladen mit =	"ABcdeF"
	Abschließende Zeichenzählung wird reduziert auf =	0x0000
	Zielblock erweitert auf niederwertiges Byte 400104	
8b	Quelle enthält =	"abCdeF12"
	Zielblock beim höherwertigen Byte 400301	
	Anfängliche Zeichenzählung =	0x0008
	Groß-/Kleinschreibung für Konvertierungssteuerung aktiviert, und die Konvertierung von Kleinschreibung in Großschreibung ist ausgewählt	
	Zielblock ist geladen mit =	"ABCDEF12"
	Abschließende Zeichenzählung wird reduziert auf =	0x0000
	Zielblock erweitert auf höherwertiges Byte 400305	

XMIT-Konvertierungsblock-Opcode – Beispiele 7 bis 11

Beschreibung des Beispiels für Opcode 7

Opcode 7 verwendet einen Quellregisterblock mit 16-Bit-Ganzzahlen und einen Zielregisterblock mit 16-Bit-Ganzzahlen. Jedes Quellwort des Quellregisterblocks wird gelesen, die Byte werden ausgetauscht und anschließend in den Zielregisterblock geschrieben. Das anfängliche Zeichenzählungsregister für ASCII-Zeichenfolgen (4x + 7) gibt die Anzahl der Register an, die konvertiert werden sollen. Dabei muss es sich um eine gerade Zahl zwischen 2 und 1024 handeln.

Beschreibung des Beispiels für Opcode 8a und 8b

Opcode 8 kopiert die ASCII-Zeichenfolge aus dem Quellregisterblock in den Zielregisterblock. Das anfängliche Zeichenzählungsregister für ASCII-Zeichenfolgen (4x + 7) gibt die Anzahl der zu kopierenden Zeichen an. Wenn die Groß- und Kleinschreibung im Steuerungsregister für die Datenkonvertierung deaktiviert ist, wird die ausgewählte Konvertierung von Klein- in Großschreibung (4x + 3, Bit 8) oder von Groß- in Kleinschreibung (4x + 3, Bit 7) während des Kopiervorgangs im Zielblock durchgeführt.
Tabelle der Konvertierungsbeispiele bezogen auf die ASCII-Zeichenfolge

Opcode	Aktionen	Daten (von Ihnen eingegeben)
9	Zielblock enthält =	"abcde"
	Quellblock enthält beim höherwertigen Byte 400201 =	"abcdefgh"
	Anfängliche Zeichenzählung =	0x0008
	Quellblock, erweitert auf das niederwertige Byte 400203, zielt jetzt auf =	"fgh"
	Abschließende Zeichenzählung wird reduziert auf =	0x0003
	Der untere Ausgang "Befehl erfolgreich" wird aktiviert, da die Zielzeichenfolge mit der Quellzeichenfolge übereinstimmt	
10	Zielblock enthält =	"def"
	Quellblock enthält beim höherwertigen Byte 400201 =	"abcdefgh"
	Anfängliche Zeichenzählung =	0x0008
	Quellblock, erweitert auf das niederwertige Byte 400202, zielt jetzt auf =	"defgh"
	Abschließende Zeichenzählung wird reduziert auf =	0x0005
	Der untere Ausgang "Befehl erfolgreich" wird aktiviert, da die Zielzeichenfolge in der Quellzeichenfolge gefunden wurde	

Opcode	Aktionen	Daten (von Ihnen eingegeben)
11	Quellblock enthält beim höherwertigen Byte 400201 =	0x0103, 0x0001, 0x0008, 0x1234
	Anfängliche Zeichenzählung =	0x0006
	Durch die Konvertierungssteuerung wird ausgewählt, dass LRC8 angehängt werden muss	
	Quellblock beim niederwertigen Byte 400201 enthält jetzt =	0x0103, 0x0001, 0x0008, 0xF334
	Quellblock bleibt beim höherwertigen Byte 400201	
	Abschließende Zeichenzählung wird erhöht auf =	0x0007
	Der untere Ausgang "Befehl erfolgreich" wird aktiviert, da die Zielzeichenfolge in der Quellzeichenfolge gefunden wurde	

Beschreibung des Beispiels für Opcode 9

Opcode 9 vergleicht die im Zielregisterblock definierte ASCII-Zeichenfolge und vergleicht sie mit dem Quellregisterblock. Die anfängliche Zeichenzählung für ASCII-Zeichenfolgen (4x + 7) gibt die maximale Anzahl der zu vergleichenden Zeichen an. Dies muss eine Zahl zwischen 1 und 1024 sein. Die Zeichenfolge zur Abgleichung ist im Zielblock enthalten und muss durch das Zeichen 0x00 abgeschlossen werden. Der Quellzeiger (4x + 3, Bit 13) wird über das letzte übereinstimmende Zeichen hinaus erweitert, und die Zeichenzählung (4x + 7) wird durch die Anzahl der übereinstimmenden Zeichen reduziert. Wenn alle Zeichen in der Quellzeichenfolge bis zum NULL-Terminator mit der Zielzeichenfolge übereinstimmen, wird der untere Ausgang eingeschaltet (Befehl erfolgreich). Ansonsten wird der mittlere Ausgang aktiviert (Fehler).

Beschreibung des Beispiels für Opcode 10

Opcode 10 durchsucht den Quellregisterblock nach der im Zielregisterblock definierten ASCII-Zeichenfolge. Die anfängliche Zeichenzählung für ASCII-Zeichenfolgen (4x + 7) gibt die maximale Anzahl der zu suchenden Zeichen an. Dies muss eine Zahl zwischen 1 und 1024 sein. Die Zeichenfolge zur Abgleichung ist im Zielblock enthalten und muss durch das Zeichen 0x00 abgeschlossen werden. Wenn eine übereinstimmende Zeichenfolge im Quellblock vorhanden ist, wird der Quellpunkt (4x + 3, Bit 13) auf den Anfang der übereinstimmenden Zeichenfolge erweitert. Die Zeichenzählung (4x + 7) wird durch die Anzahl der Zeichen reduziert, die am Anfang des Quellblocks übersprungen wurden, und der untere Ausgang wird aktiviert (Befehl erfolgreich). Anderenfalls werden der Quellzeiger und die Zeichenzählung nicht geändert, und der mittlere Ausgang wird aktiviert (Fehler).

Beschreibung des Beispiels für Opcode 11

Opcode 11 führt eine Fehlerprüfungsberechnung für 8-Bit-LRC, 16-Bit-CRC mit Seed 0xFFFF oder 16-Bit-CRC mit Seed 0x0000 durch. Wenn das Konvertierungssteuerungsregister (4x +3, Bit 4) eingestellt ist (validieren), wird die ausgewählte Fehlerprüfung am Ende der ASCII-Zeichenfolge im Quellblock validiert, wobei die vorgegebene Länge durch die Zeichenzählung der ASCII-Zeichenfolge bestimmt wird. Wenn die Fehlerprüfung gültig ist, wird der untere Ausgang aktiviert (Befehl erfolgreich). Ansonsten wird der mittlere Ausgang aktiviert (Fehler).
Wenn das Konvertierungssteuerungsregister (4x +3, Bit 4) 0 ist (anhängen), wird die ausgewählte Fehlerprüfung berechnet und an das Ende der ASCII-Zeichenfolge im Quellblock angehängt. Die Zeichenzählung wird durch die Byte-Größe der Fehlerprüfung erhöht, der Quellzeiger wird nicht erweitert, und der untere Ausgang wird aktiviert (Befehl erfolgreich).

XMIT Binär/BCD-Konvertierungstypen

Konvertierung von binär nach BCD

Um diesen Konvertierungstyp auszuführen, ist die Verwendung von zwei XMIT-Konvertierungsblöcken erforderlich. Der erste XMIT-Konvertierungsblock verwendet (Opcode 4) für die Konvertierung der Ganzzahl der 32-Bit-Binärquelle in eine feste aus 10 Zeichen bestehende ASCII-Dezimalzeichenfolge, die in einem 4x-Registerblock gespeichert wird. Der zweite XMIT-Konvertierungsblock verwendet (Opcode 2) für die Konvertierung einer ASCII-Zeichenfolge im Hexadezimalformat, die aus demselben 4x-Registerblock abgelesen wird, in eine 32-Bit-BCD-Ganzzahl. Die Ganzzahl der Binärquelle muss kleiner sein als 0x05F5E0FF, was 99999999 im Dezimalformat entspricht.

Konvertierung von BCD nach binär

Um diesen Konvertierungstyp auszuführen, ist die Verwendung von zwei XMIT-Konvertierungsblöcken erforderlich. Der erste XMIT-Konvertierungsblock verwendet (Opcode 5) für die Konvertierung der Ganzzahl der 32-Bit-BCD-Quelle in eine feste aus 8 Ziffern bestehende ASCII-Hexadezimalzeichenfolge, die in einem 4x-Registerblock gespeichert wird. Der zweite XMIT-Konvertierungsblock verwendet (Opcode 1) für die Konvertierung einer ASCII-Zeichenfolge im Dezimalformat, die aus demselben 4x-Registerblock abgelesen wird, in eine binäre 32-Bit-Ganzzahl. Wenn alle 8 Zeichen analysiert werden, ist die BCD-Zielganzzahl eine gültige BCD-Zahl.

Arbeiten mit XMIT-Beispielen

5

Auf einen Blick

Zweck

In diesem Kapitel werden XMIT-Anwendungsbeispiele beschrieben.

Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Abschnitte:

Abschnitt	Thema	Seite
5.1	Einfache ASCII-Lese-/-Schreibvorgänge und Modbus-Lese-/-Schreibvorgänge	134
5.2	Übertragen mehrerer Modbus-Befehle: SPS-Master an SPS-Slave	151
5.3	Übertragung des Fehlerworts über Impulswahlmodems an den SPS-Slave	162

5.1 Einfache ASCII-Lese-/-Schreibvorgänge und Modbus-Lese-/-Schreibvorgänge

Auf einen Blick

Zweck

Schneider Electric hat die Netzwerk-Programmierung mit der Steuerungs-Software Modsoft eingeführt. Jetzt bietet Schneider Electric drei PC-basierte GUI-Anwendungen zur Netzwerk-Programmierung an:

- Concept
- ProWORX NxT
- ProWORX32

Verwenden Sie die Steuerungs-Software entweder zur Konfiguration von Registern oder zur Anzeige der Daten in einem Register (Wert).
In diesem Abschnitt finden Sie Beispiele für einfache ASCII- bzw. Modbus-Lese- und Schreibvorgänge unter Verwendung der Steuerungs-Software Concept, ProWORX32 oder ProWORX NxT. Jede der Anwendungen bietet eine andere Benutzerschnittstelle für die Konfiguration und Anzeige.

Inhalt dieses Abschnitts

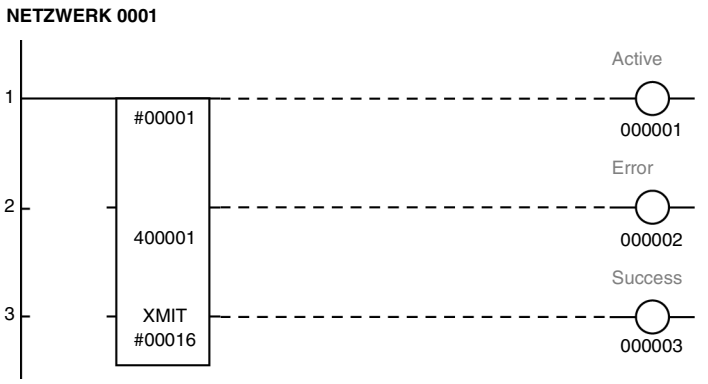
Dieser Abschnitt enthält die folgenden Themen:

Thema	Seite
Lesen von einfachen ASCII-Zeichen mit Hilfe von Concept	135
Schreiben von einfachen ASCII-Zeichen mit Hilfe von ProWORX32	139
Modbus-Lesevorgänge, bei denen ProWORX NxT verwendet wird	143
Modbus-Schreibvorgänge an RS485-Schnittstelle Nr. 2 unter Verwendung von ProWORX NxT	147

Lesen von einfachen ASCII-Zeichen mit Hilfe von Concept

Inhalt des Eintrags

Netzwerkdiagramm



Inhalt des Eintrags

- Oberer Eintrag: enthält #00001, um die Kommunikation des XMIT-Blocks über die Schnittstelle Nr. 1 der Master-SPS zu leiten
- Mittlerer Eintrag: 400001 ist das Anfangsregister, das zur Konfiguration des XMIT-Blocks verwendet wird
- Unterer Eintrag: Auf #000016 gesetzt

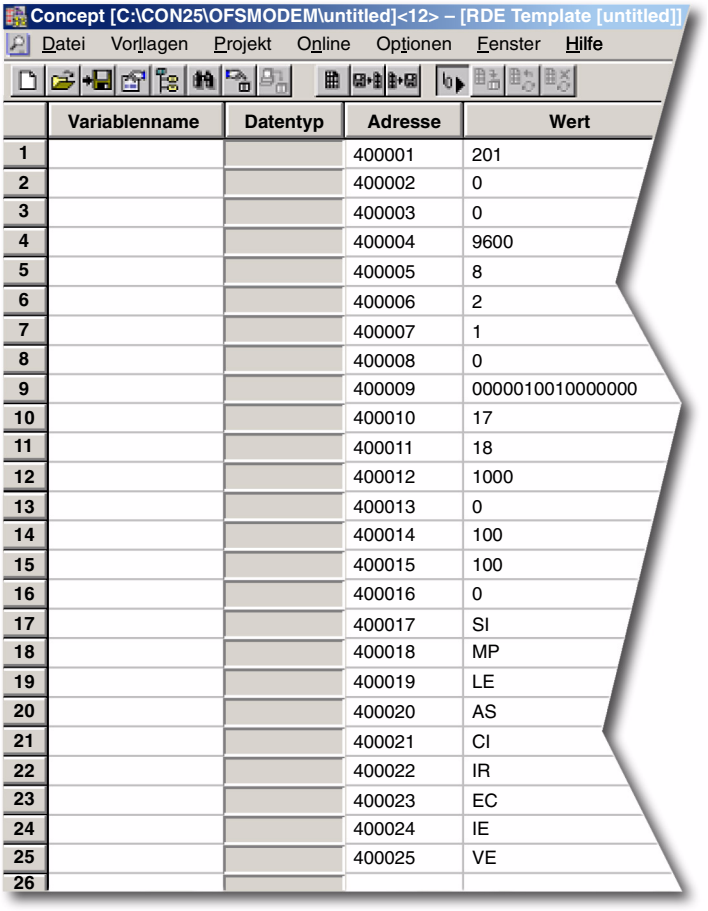
  Der untere Eintrag muss auf #000016 gesetzt werden, wenn ein ASCII-Lese- oder Schreibvorgang ausgeführt wird. Die Modbus-Funktionscodes müssen 1 bis 6, 15 und 16 lauten.

Die drei rechten Spulen geben den Status des XMIT-Blocks an.

Hinweis: Wenn Sie mehrere XMIT-Blöcke verwenden oder dieselben Blöcke für aufeinander folgende Übertragungen auf eines oder mehrere Slave-Geräte verwenden, können diese Ausgänge verwendet werden, um die XMIT-Blöcke zu aktivieren. Zu einem bestimmten Zeitpunkt ist jeweils nur eine Modbus-Meldung über eine bestimmte serielle Schnittstelle zulässig.

Konfiguration und Anzeige von Daten mit Concept

Referenzdaten-Editor (RDE)



	Variablenname	Datentyp	Adresse	Wert
1			400001	201
2			400002	0
3			400003	0
4			400004	9600
5			400005	8
6			400006	2
7			400007	1
8			400008	0
9			400009	0000010010000000
10			400010	17
11			400011	18
12			400012	1000
13			400013	0
14			400014	100
15			400015	100
16			400016	0
17			400017	SI
18			400018	MP
19			400019	LE
20			400020	AS
21			400021	CI
22			400022	IR
23			400023	EC
24			400024	IE
25			400025	VE
26				

Um das Zoomfenster aufzurufen, platzieren Sie den Cursor auf dem XMIT-Block, und drücken Sie "Strg+D".

Interpretieren der Daten im RDE

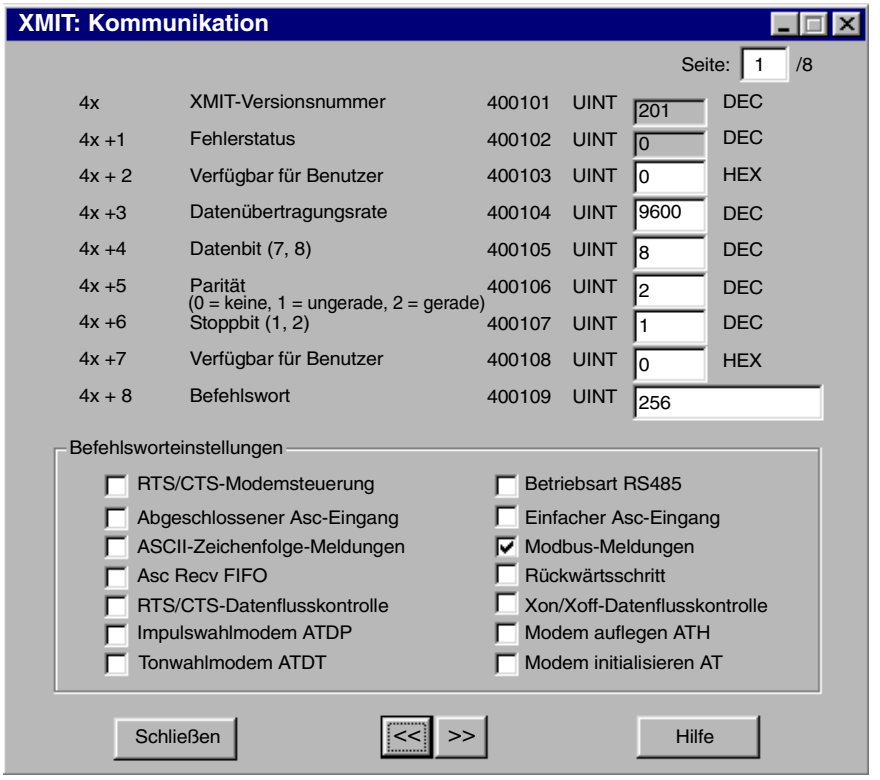
Die Daten enthalten die folgenden Informationen:

- Halteregister 4x + 8 (Befehlswort, Adresse 40009) setzt Bit sechs (6) (Einfachen ASCII-Eingang aktivieren/deaktivieren) und Bit neun (9) (ASCII-Empfangs-FIFO aktivieren/deaktivieren)
- Halteregister 4x + 9 (Nachrichtenzeiger, Adresse 40010) weist einen Wert von 17 auf und ist ein Zeiger auf den ASCII-Text, der gelesen wird. Der Wert wird in Form von zwei Zeichen pro Register, beginnend bei Adresse 400017, gespeichert.
- Halteregister 4x + 10 (Nachrichtenlänge, Adresse 40011) wird auf 18 gesetzt, da die gelesene ASCII-Zeichenfolge 18 Zeichen lang ist

Der an Adresse 400017–40025 angezeigte ASCII-Text wurde über eine PC-Tastatur mit einer geöffneten Microsoft HyperTerminal Com-Schnittstellenverbindung eingegeben.

Register-funktionen

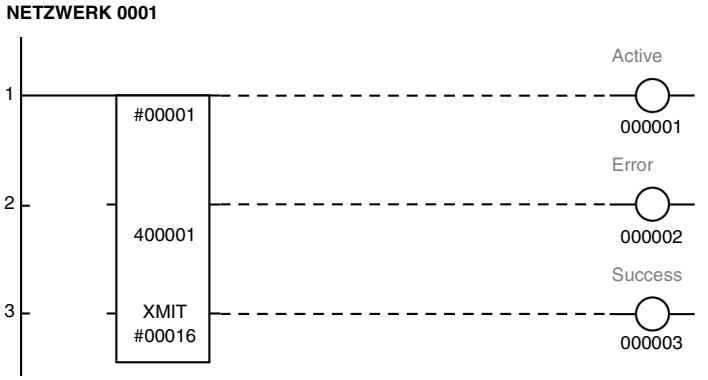
In einem Zoomfenster angezeigter Registerinhalt.



Schreiben von einfachen ASCII-Zeichen mit Hilfe von ProWORX32

Inhalt des Eintrags

Netzwerkdiagramm



Inhalt des Eintrags

- Oberer Eintrag: enthält #00001, um die Kommunikation des XMIT-Blocks über die Schnittstelle Nr. 1 der Master-SPS zu leiten
- Mittlerer Eintrag: 400001 ist das Anfangsregister, das zur Konfiguration des XMIT-Blocks verwendet wird
- Unterer Eintrag: Auf #000016 gesetzt

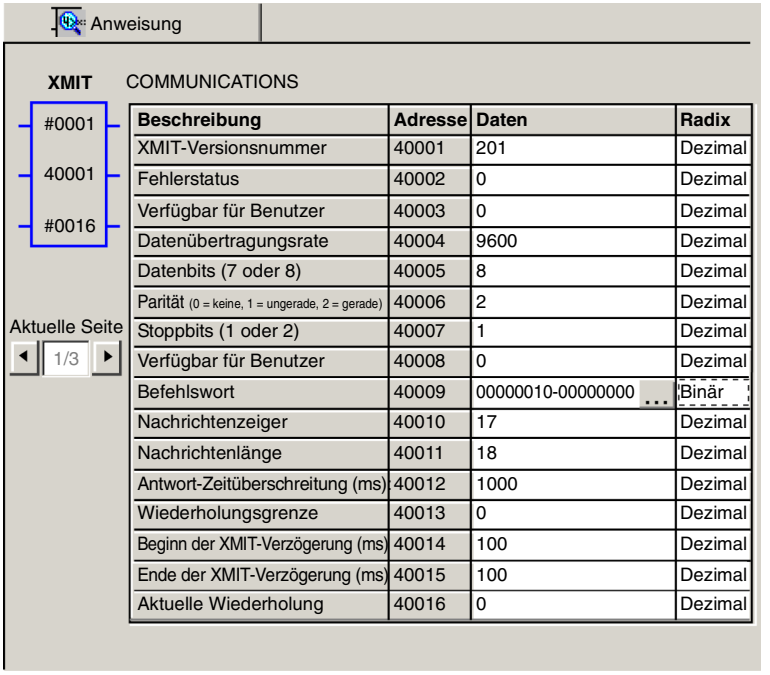
  Der untere Eintrag muss auf #000016 gesetzt werden, wenn ein ASCII-Lese- oder -Schreibvorgang ausgeführt wird. Die Modbus-Funktionscodes müssen 1 bis 6, 15 und 16 lauten.

Die drei rechten Spulen geben den Status des XMIT-Blocks an.

Hinweis: Wenn Sie mehrere XMIT-Blöcke verwenden oder dieselben Blöcke für aufeinander folgende Übertragungen auf eines oder mehrere Slave-Geräte verwenden, können diese Ausgänge verwendet werden, um die XMIT-Blöcke zu aktivieren. Zu einem bestimmten Zeitpunkt ist jeweils nur eine Modbus-Meldung über eine bestimmte serielle Schnittstelle zulässig.

Konfiguration und Anzeige von Daten in ProWORX32

Zoomfenster für die Kommunikation



Beschreibung	Adresse	Daten	Radix
XMIT-Versionsnummer	40001	201	Dezimal
Fehlerstatus	40002	0	Dezimal
Verfügbar für Benutzer	40003	0	Dezimal
Datenübertragungsrate	40004	9600	Dezimal
Datenbits (7 oder 8)	40005	8	Dezimal
Parität (0 = keine, 1 = ungerade, 2 = gerade)	40006	2	Dezimal
Stoppbits (1 oder 2)	40007	1	Dezimal
Verfügbar für Benutzer	40008	0	Dezimal
Befehlswort	40009	00000010-00000000 ...	Binär
Nachrichtenzeiger	40010	17	Dezimal
Nachrichtenlänge	40011	18	Dezimal
Antwort-Zeitüberschreitung (ms)	40012	1000	Dezimal
Wiederholungsgrenze	40013	0	Dezimal
Beginn der XMIT-Verzögerung (ms)	40014	100	Dezimal
Ende der XMIT-Verzögerung (ms)	40015	100	Dezimal
Aktuelle Wiederholung	40016	0	Dezimal

Interpretation des Zoomfensters für die Kommunikation

Die Daten enthalten die folgenden Informationen:

- Halteregister 4x + 8 (Befehlswort, Adresse 40009) legt Bit fest. Bit 7 aktiviert bzw. deaktiviert beispielsweise die Nachrichtenübermittlung mit einfachen ASCII-Zeichenfolgen.

  Siehe nachfolgendes Dialogfeld zur Darstellung der Bit-Anzeige.

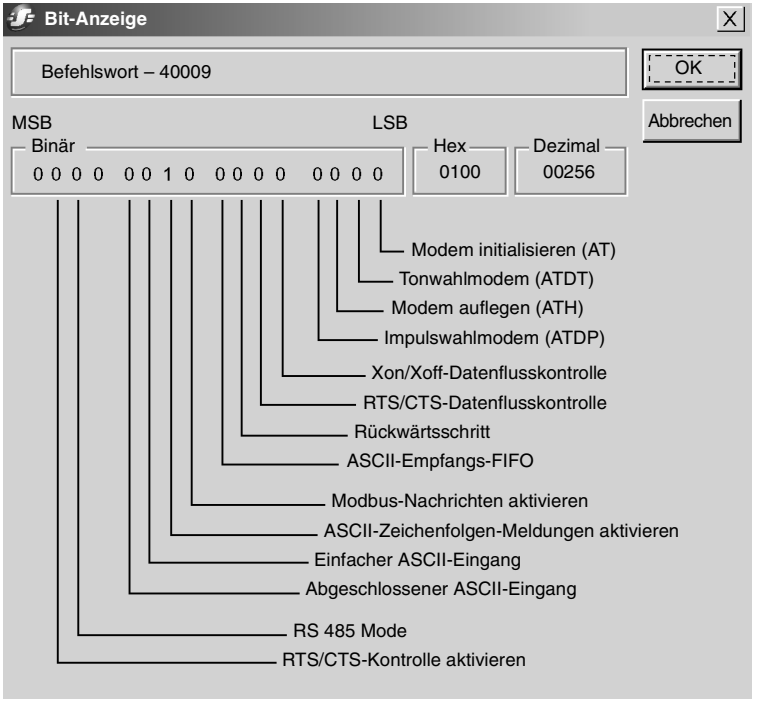
- Halteregister 4x + 9 (Nachrichtenzeiger, Adresse 40010) weist einen Wert von 17 auf und ist ein Zeiger auf den ASCII-Text, der geschrieben wird. Der Wert wird in Form von zwei Zeichen pro Register, beginnend bei Adresse 400017, gespeichert.

  Siehe nachfolgendes Dialogfeld zur Datenanzeige.

- Halteregister 4x + 10 (Nachrichtenlänge, Adresse 40011) wird auf 18 gesetzt, da die abgelesene ASCII-Zeichenfolge 18 Zeichen lang ist.

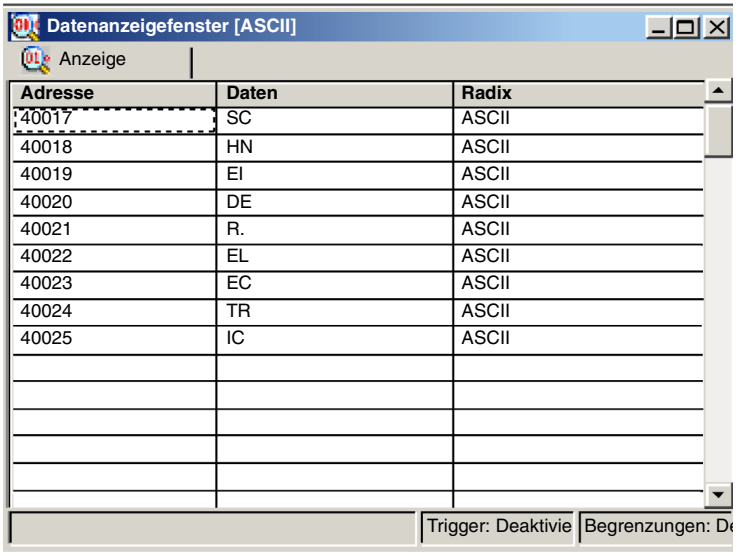
Aufrufen des Dialogfelds "Bit-Anzeige"

Bit 7 ist auf 1 gesetzt. Dadurch wird die Nachrichtenübermittlung mit ASCII-Zeichenketten aktiviert.



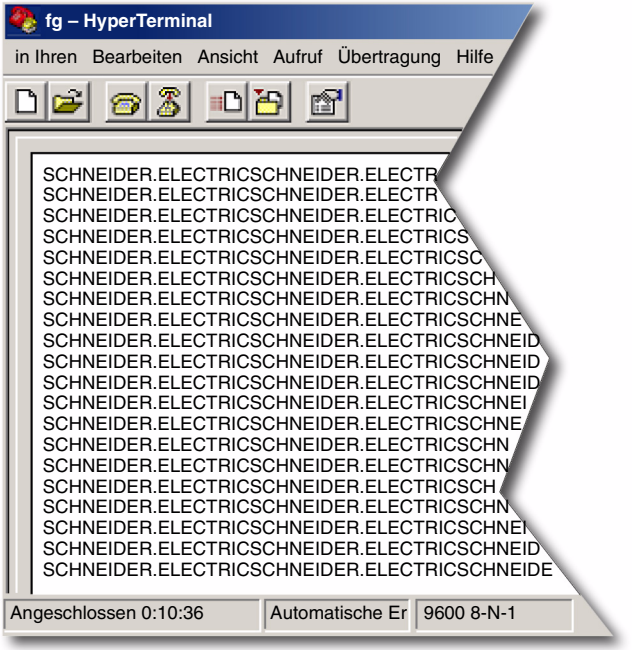
Darstellen von Daten im Datenanzeigefenster

Inhalt der Register 40017 bis 40025. Vom Benutzer eingegebener Inhalt.



Verwendung von Microsoft HyperTerminal zur Anzeige von Daten

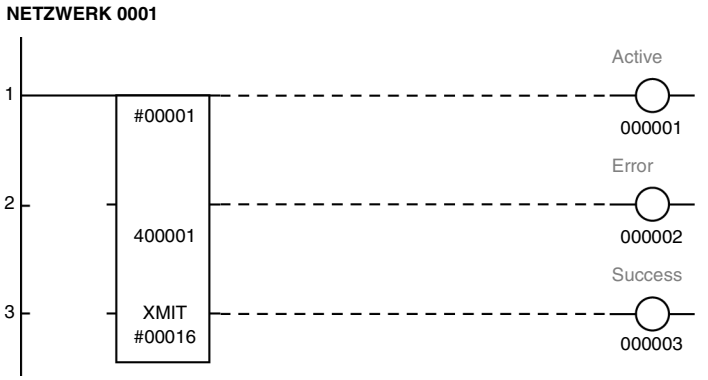
Der Inhalt der Register 40017 bis 40025 wird als fortlaufende Textzeichenfolge angezeigt



Modbus-Lesevorgänge, bei denen ProWORX NxT verwendet wird

Inhalt des Eintrags

Netzwerkdiagramm



Inhalt des Eintrags

- Oberer Eintrag: enthält #00001, um die Kommunikation des XMIT-Blocks über die Schnittstelle Nr. 1 der Master-SPS zu leiten
- Mittlerer Eintrag: 400001 ist das Anfangsregister, das zur Konfiguration des XMIT-Blocks verwendet wird
- Unterer Eintrag: Auf #000016 gesetzt

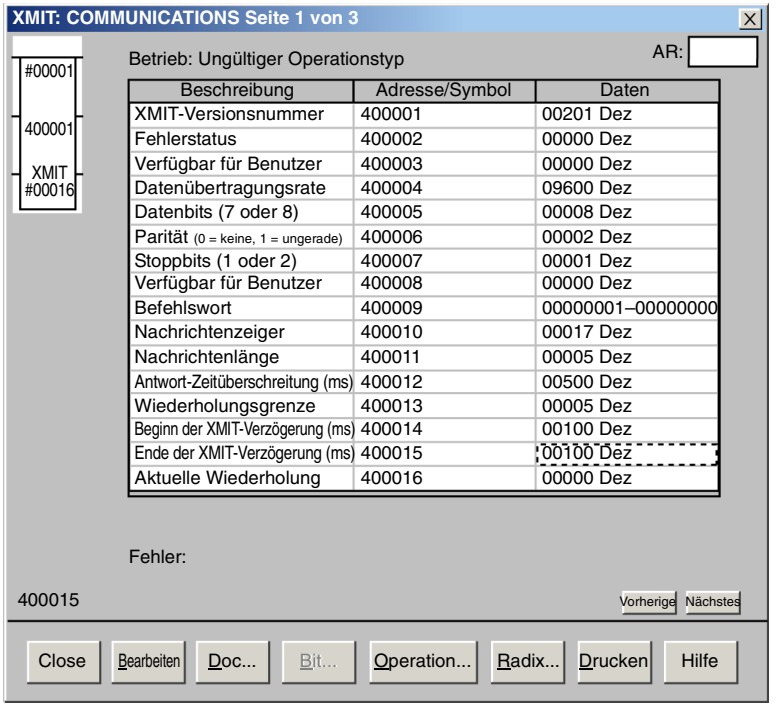
  Der untere Eintrag muss auf #000016 gesetzt werden, wenn ein ASCII-Lese- oder -Schreibvorgang ausgeführt wird. Die Modbus-Funktionscodes müssen 1 bis 6, 15 und 16 lauten.

Die drei rechten Spulen geben den Status des XMIT-Blocks an.

Hinweis: Wenn Sie mehrere XMIT-Blöcke verwenden oder dieselben Blöcke für aufeinander folgende Übertragungen auf eines oder mehrere Slave-Geräte verwenden, können diese Ausgänge verwendet werden, um die XMIT-Blöcke zu aktivieren. Zu einem bestimmten Zeitpunkt ist jeweils nur eine Modbus-Meldung über eine bestimmte serielle Schnittstelle zulässig.

Anzeige der Daten in ProWORX NxT

Platzieren Sie den Cursor auf dem XMIT-Block, und drücken Sie "Strg+R", um dieses Zoomfenster anzuzeigen.


XMIT: COMMUNICATIONS Seite 1 von 3
#00001
400001
XMIT
#00016
Betrieb: Ungültiger Operationstyp
AR:
Beschreibung | Adresse/Symbol | Daten
XMIT-Versionsnummer | 400001 | 00201 Dez
Fehlerstatus | 400002 | 00000 Dez
Verfügbar für Benutzer | 400003 | 00000 Dez
Datenübertragungsrate | 400004 | 09600 Dez
Datenbits (7 oder 8) | 400005 | 00008 Dez
Parität (0 = keine, 1 = ungerade) | 400006 | 00002 Dez
Stoppbits (1 oder 2) | 400007 | 00001 Dez
Verfügbar für Benutzer | 400008 | 00000 Dez
Befehlswort | 400009 | 00000001–00000000
Nachrichtenzeiger | 400010 | 00017 Dez
Nachrichtenlänge | 400011 | 00005 Dez
Antwort-Zeitüberschreitung (ms) | 400012 | 00500 Dez
Wiederholungsgrenze | 400013 | 00005 Dez
Beginn der XMIT-Verzögerung (ms) | 400014 | 00100 Dez
Ende der XMIT-Verzögerung (ms) | 400015 | 00100 Dez
Aktuelle Wiederholung | 400016 | 00000 Dez
Fehler:
400015
Vorherige
Nächstes
Close
Bearbeiten
Doc...
Bit...
Operation...
Radix...
Drucken
Hilfe

Inhalt des Registers

Inhalt des Registers

Register	Zugriff	Beschreibung	Registerwert
4x	Lesen	Überprüfung des XMIT-Blocks	
4x + 1	Lesen	Fehlerstatus	
4x + 2	Schreiben	Verfügbar für Benutzer Kann als Zeiger verwendet werden	
4x + 3	Schreiben	Baudrate 50, 75, 110, 134, 150, 300, 600, 1200, 2400, 9600 und 19200	9600
4x + 4	Schreiben	Datenbit 7, 8	8
4x + 5	Schreiben	Parität 0 = keine; 1 = ungerade; 2 = gerade	2
4x + 6	Schreiben	Stoppbit 0, 1, 2	1
4x + 7	Schreiben	Verfügbar für Benutzer Kann als Zeiger verwendet werden	
4x + 8	Schreiben	Befehlswort 0000-0000-0000-0000	256 dezimal
4x + 9	Schreiben	Zeiger Offset für ein 4x-Halteregister zur weiteren Konfiguration des XMIT-Blocks	17
4x + 10	Schreiben	Länge Für Modbus-Nachrichtenübermittlung Muss auf 5 gesetzt werden	5
4x + 11	Schreiben	Antwort-Timeout $_{ms}$	500
4x + 12	Schreiben	Wiederholungsgrenze	5
4x + 13	Schreiben	Beginn der Verzögerung $_{ms}$	100
4x + 14	Schreiben	Ende der Verzögerung $_{ms}$	100
4x + 15	Lesen	Aktuelle Wiederholung	

Interpretieren der Daten

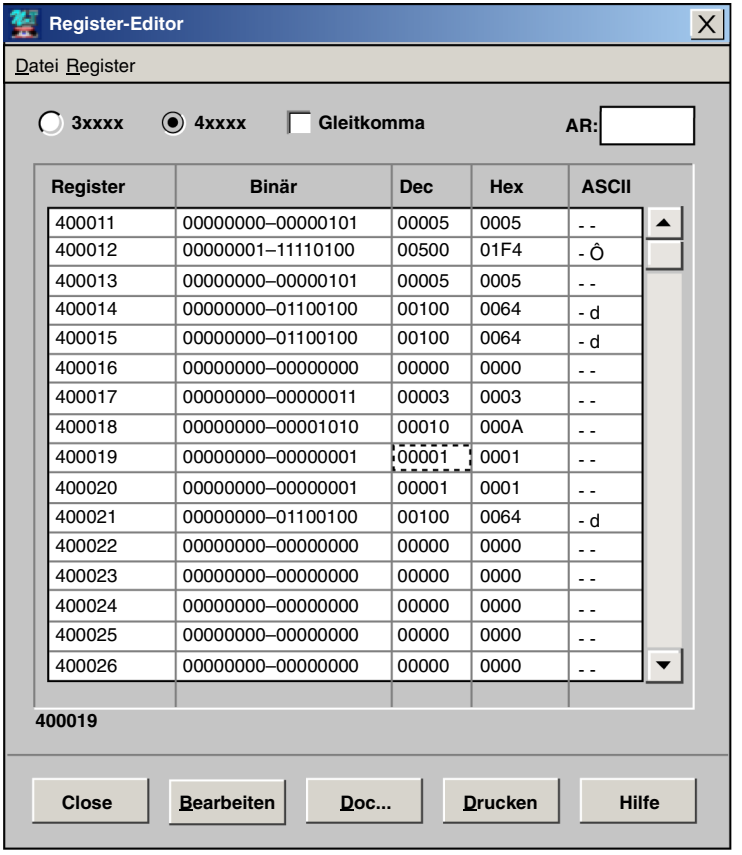
Register 4x + 9 verweist auf einen Zeigerwert.

Hinweis: Konfiguration eines XMIT-Blocks für Modbus-Meldungen
Konfigurieren Sie fünf zusätzliche aufeinander folgende Register, um einen XMIT-Block für Modbus-Meldungen funktionsfähig zu machen.

In diesem Beispiel verfügt das Register 4x + 9 über den Wert 17. Daher beginnt die Konfiguration für die Modbus-Meldung bei 4x + 17.

Verwendung des Zoomfensters für den Register-Editor

Register-Editor



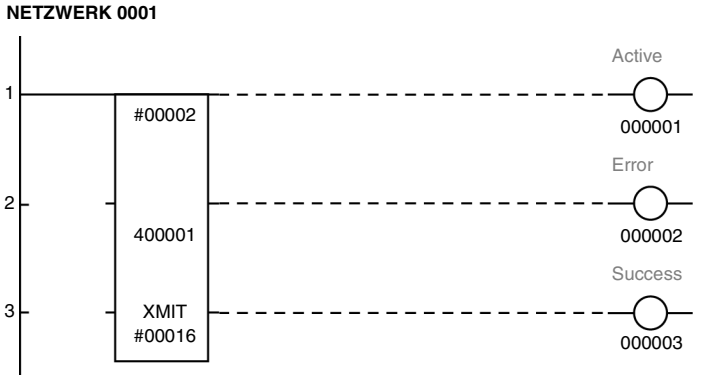
Inhalt des Registers

Register	Zugriff	Beschreibung	Registerwert
4x 17	Lesen	Modbus-Funktionscode	3
4x 18	Schreiben	Anzahl der zu übertragenden Register	10
4x 19	Schreiben	Adresse des Modbus-Slave-Geräts	1
4x 20	Schreiben	Slave-Adresse	1 = 400001, vom Slave gelesene Daten
4x 21	Schreiben	Offset des Datenbereichs der Master-SPS	100 = 40010, im Master platzierte Daten

Modbus-Schreibvorgänge an RS485-Schnittstelle Nr. 2 unter Verwendung von ProWORX NxT

Inhalt des Eintrags

Netzwerkdiagramm



Inhalt des Eintrags

- Oberer Eintrag: enthält #00002, um die Kommunikation des XMIT-Blocks über die Schnittstelle Nr. 2 der Master-SPS zu leiten
- Mittlerer Eintrag: 400001 ist das Anfangsregister, das zur Konfiguration des XMIT-Blocks verwendet wird
- Unterer Eintrag: Auf #000016 gesetzt

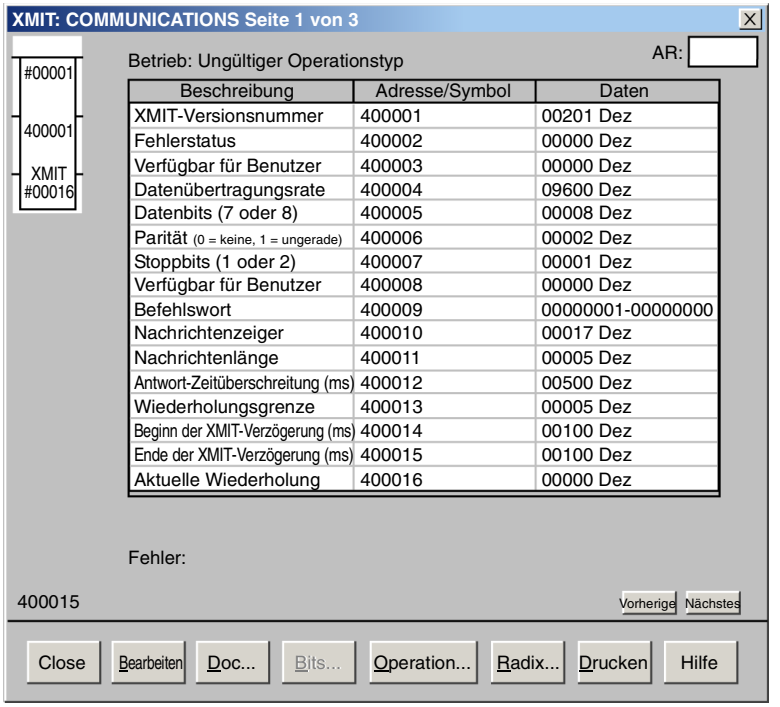
  Der untere Eintrag muss auf #000016 gesetzt werden, wenn ein ASCII-Lese- oder -Schreibvorgang ausgeführt wird. Die Modbus-Funktionscodes müssen 1 bis 6, 15 und 16 lauten.

Die drei rechten Spulen geben den Status des XMIT-Blocks an.

Hinweis: Wenn Sie mehrere XMIT-Blöcke verwenden oder dieselben Blöcke für aufeinander folgende Übertragungen auf eines oder mehrere Slave-Geräte verwenden, können diese Ausgänge verwendet werden, um die XMIT-Blöcke zu aktivieren. Zu einem bestimmten Zeitpunkt ist jeweils nur eine Modbus-Meldung über eine bestimmte serielle Schnittstelle zulässig.

Anzeige der Daten in ProWORX NxT

Platzieren Sie den Cursor auf dem XMIT-Block, und drücken Sie "Strg+R", um dieses Zoomfenster anzuzeigen.


XMIT: COMMUNICATIONS Seite 1 von 3
#00001
400001
XMIT
#00016
Betrieb: Ungültiger Operationstyp
AR:
Beschreibung | Adresse/Symbol | Daten
XMIT-Versionsnummer | 400001 | 00201 Dez
Fehlerstatus | 400002 | 00000 Dez
Verfügbar für Benutzer | 400003 | 00000 Dez
Datenübertragungsrate | 400004 | 09600 Dez
Datenbits (7 oder 8) | 400005 | 00008 Dez
Parität (0 = keine, 1 = ungerade) | 400006 | 00002 Dez
Stoppbits (1 oder 2) | 400007 | 00001 Dez
Verfügbar für Benutzer | 400008 | 00000 Dez
Befehlswort | 400009 | 00000001-00000000
Nachrichtenzeiger | 400010 | 00017 Dez
Nachrichtenlänge | 400011 | 00005 Dez
Antwort-Zeitüberschreitung (ms) | 400012 | 00500 Dez
Wiederholungsgrenze | 400013 | 00005 Dez
Beginn der XMIT-Verzögerung (ms) | 400014 | 00100 Dez
Ende der XMIT-Verzögerung (ms) | 400015 | 00100 Dez
Aktuelle Wiederholung | 400016 | 00000 Dez
Fehler:
400015
Vorherige
Nächstes
Close
Bearbeiten
Doc...
Bits...
Operation...
Radix...
Drucken
Hilfe

Inhalt des Registers

Inhalt des Registers

Register	Zugriff	Beschreibung	Registerwert
4x	Lesen	Überprüfung des XMIT-Blocks	
4x + 1	Lesen	Fehlerstatus	
4x + 2	Schreiben	Verfügbar für Benutzer Kann als Zeiger verwendet werden	
4x + 3	Schreiben	Baudrate 50, 75, 110, 134, 150, 300, 600, 1200, 2400, 9600 und 19200	9600
4x + 4	Schreiben	Datenbit 7, 8	8
4x + 5	Schreiben	Parität 0 = keine; 1 = ungerade; 2 = gerade	2
4x + 6	Schreiben	Stoppbit 0, 1, 2	1
4x + 7	Schreiben	Verfügbar für Benutzer Kann als Zeiger verwendet werden	
4x + 8	Schreiben	Befehlswort 0010-0001-0000-0000	8448 dezimal 2100 Hex
4x + 9	Schreiben	Zeiger Offset für ein 4x-Halteregister zur weiteren Konfiguration des XMIT-Blocks	17
4x + 10	Schreiben	Länge Für Modbus-Nachrichtenübermittlung Muss auf 5 gesetzt werden	5
4x + 11	Schreiben	Antwort-Timeout $_{ms}$	500
4x + 12	Schreiben	Wiederholungsgrenze	5
4x + 13	Schreiben	Beginn der Verzögerung $_{ms}$	100
4x + 14	Schreiben	Ende der Verzögerung $_{ms}$	100
4x + 15	Lesen	Aktuelle Wiederholung	

Interpretieren der Daten

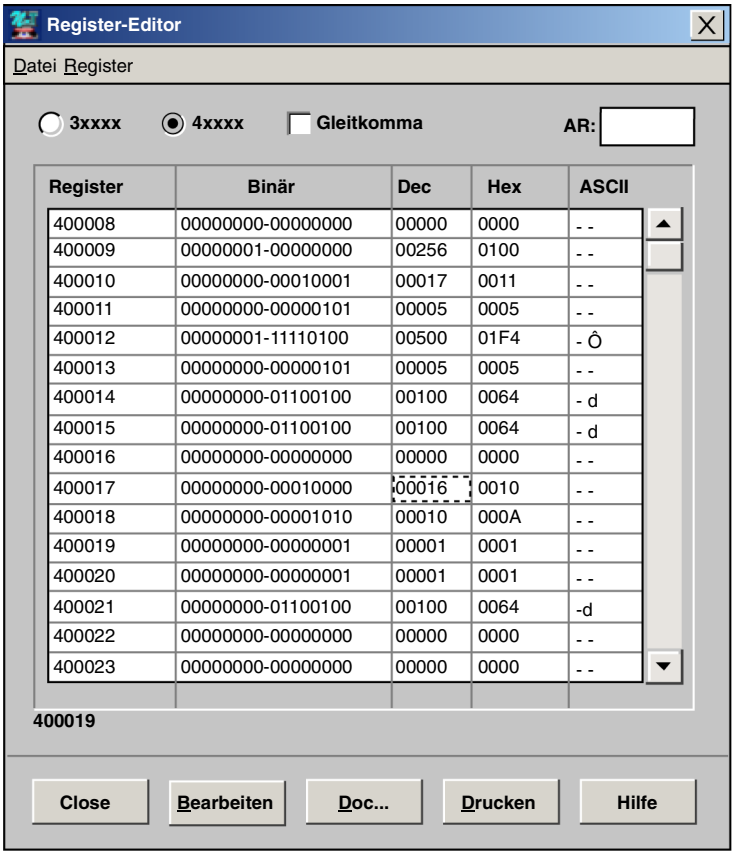
Register 4x + 9 ist ein Zeiger auf einen Satz von fünf benachbarten Konfigurationsregistern.

Hinweis: Konfiguration eines XMIT-Blocks für Modbus-Meldungen
Konfigurieren Sie fünf zusätzliche aufeinander folgende Register, um einen XMIT-Block für Modbus-Meldungen funktionsfähig zu machen.

In diesem Beispiel verfügt das Register 4x + 9 über den Wert 17. Daher beginnt die Konfiguration für die Modbus-Meldung bei 4x + 17.

Verwendung des Zoomfensters für den Register-Editor

Register-Editor



Inhalt des Registers

Register	Zugriff	Beschreibung	Registerwert
4x 17	Schreiben	Modbus-Funktionscode	16
4x 18	Schreiben	Anzahl der zu übertragenden Register	10
4x 19	Schreiben	Adresse des Modbus-Slave-Geräts	1
4x 20	Schreiben	Offset des Datenbereichs der Slave-SPS	1 = 400001, erstes Register im Slave-Gerät, in das Daten geschrieben werden
4x 21	Schreiben	Offset des Datenbereichs der Master-SPS	100 = 40010, das erste Register des Masters, aus dem Daten geschrieben werden

5.2 Übertragen mehrerer Modbus-Befehle: SPS-Master an SPS-Slave

Auf einen Blick

Zweck

In diesem Abschnitt wird die Übertragung mehrerer Modbus-Befehle beschrieben.

Inhalt dieses Abschnitts

Dieser Abschnitt enthält die folgenden Themen:

Thema	Seite
Senden mehrerer Modbus-Befehle	152
Einrichten der Master-SPS	153
Verwendung der Ladder Logic für mehrere Modbus-Befehle – Netzwerk Nr. 1	154
Verwendung der Ladder Logic für mehrere Modbus-Befehle – Netzwerk Nr. 2	155
Verwendung der Ladder Logic für mehrere Modbus-Befehle – Netzwerk Nr. 3	158
Verwendung der Ladder Logic für mehrere Modbus-Befehle – Netzwerk Nr. 4	160
Abschließende Übertragung mehrerer Modbus-Befehle	161

Senden mehrerer Modbus-Befehle

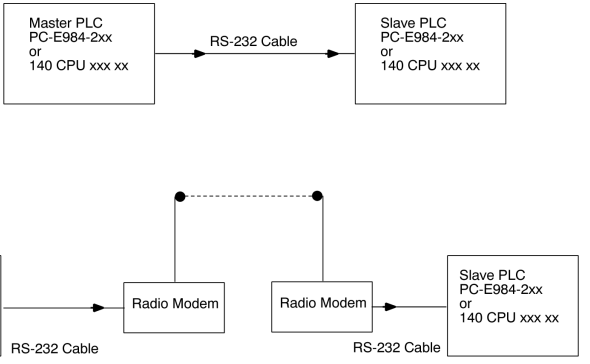
Anwendungsbeispiel unter Verwendung eines XMIT-Blocks

Die Ladder Logic in diesem Beispiel zeigt, wie mehrere Modbus-Befehle an eine Slave-SPS unter Verwendung eines einzelnen XMIT-Blocks gesendet werden. In diesem Beispiel werden vier Modbus-Befehle an eine Slave-SPS mit Modbus-Adresse Nr. 3 gesendet. Die Befehle führen folgende Funktionen aus:

- Lesen von 25 Halteregistern (4x), beginnend bei 40010 in der Slave-SPS, und Platzierung dieser Halteregister in der Master-SPS, beginnend bei 40800.
- Schreiben von 25 Halteregistern (4x), beginnend bei 40825 in der Master-SPS, und Platzierung dieser Halteregister in der Slave-SPS, beginnend bei 40010.
- Lesen von 16 Spulen (0x), beginnend bei 00001 in der Slave-SPS, und Platzierung dieser Spulen in der Master-SPS, beginnend bei 00097.
- Schreiben von 16 Spulen (0x), beginnend bei 00113 in der Master-SPS, und Platzierung dieser Spulen in der Slave-SPS, beginnend bei 00001.

Hardware konfigurieren

Siehe Abbildung zur Hardware-Konfiguration der Master-zu-Slave-SPS-Anwendung.



Hinweis: Diese Anwendung funktioniert sowohl mit Radio-Modems als auch mit Mietleitungs-Modems.

Einrichten der Master-SPS

Setup der Master-SPS

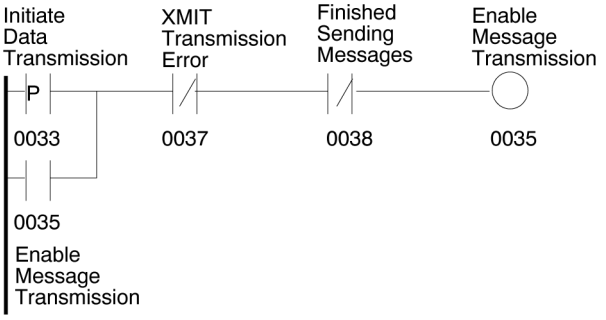
XMIT muss vier Meldungen von der Schnittstelle Nr. 1 der Master-SPS übertragen. XMIT bildet diese Meldungen anhand der vier Modbus-Meldungsdefinitionstabellen, die in der Master-SPS definiert sind. Die Modbus-Definitionstabellen sind nachfolgend dargestellt.
Siehe Tabelle mit den Modbus-Meldungsdefinitionen.

Definitionstabelle Nr. 1	Beschreibung	Register	Inhalt
	Modbus-Funktionscode	40100	3
	Menge	40101	25
	Slave-SPS-Adresse	40102	3
	Slave-SPS-Datenbereich	40103	10
	Master-SPS-Datenbereich	40104	800
Definitionstabelle Nr. 2	Beschreibung	Register	Inhalt
	Modbus-Funktionscode	40105	16
	Menge	40106	25
	Slave-SPS-Adresse	40107	3
	Slave-SPS-Datenbereich	40108	10
	Master-SPS-Datenbereich	40109	825
Definitionstabelle Nr. 3	Beschreibung	Register	Inhalt
	Modbus-Funktionscode	40110	1
	Menge	40111	16
	Slave-SPS-Adresse	40112	3
	Slave-SPS-Datenbereich	40113	1
	Master-SPS-Datenbereich	40114	97
Definitionstabelle Nr. 4	Beschreibung	Register	Inhalt
	Modbus-Funktionscode	40115	15
	Menge	40116	16
	Slave-SPS-Adresse	40117	3
	Slave-SPS-Datenbereich	40118	1
	Master-SPS-Datenbereich	40119	113

Verwendung der Ladder Logic für mehrere Modbus-Befehle – Netzwerk Nr. 1

Ladder Logic

Netzwerk Nr. 1 sendet die Modbus-Befehle an die Slave-SPS. Die Verweise auf Halteregister, Spulen und Eingänge können entsprechend Ihrer Anwendung geändert werden.
Siehe Abbildung mit den an die Slave-SPS gesendeten Modbus-Befehlen für Netzwerk Nr. 1.

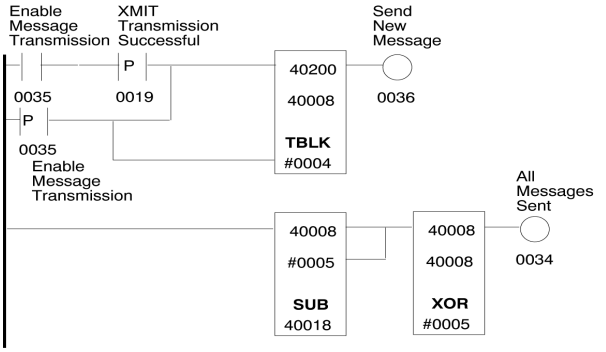


Netzwerk Nr. 1 initialisiert die Modbus-Befehle an der Slave-SPS, wenn Spule 00033 auf ON geschaltet wird. Spule 00035 bleibt so lange im Zustand ON, bis alle vier Modbus-Befehle an die Slave-SPS gesendet wurden. Wenn während einer Modbus-Übertragung an die Slave-SPS ein XMIT-Fehler auftritt, wird Spule 00035 entriegelt.

Verwendung der Ladder Logic für mehrere Modbus-Befehle – Netzwerk Nr. 2

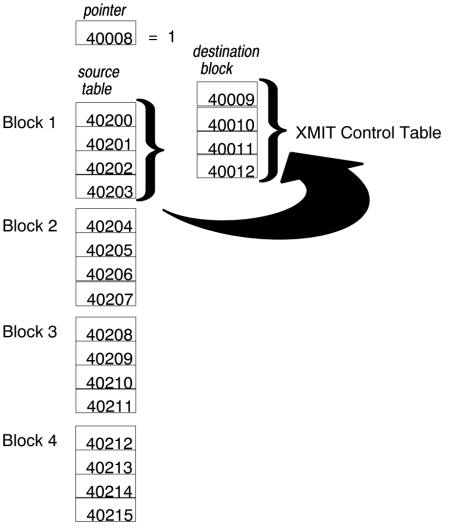
Netzwerk Nr. 2

Netzwerk Nr. 2 richtet die XMIT-Steuerungstabellendaten (40001 bis 40015) für eine neue Meldung ein.
Siehe Abbildung zur Einrichtung der XMIT-Steuerungstabelle für Netzwerk Nr. 2.



Zwei Register (4x + 2 und 4x + 7) innerhalb der XMIT-Steuerungstabelle (Registerlänge 15) sind als "Verfügbar für Benutzer" festgelegt, so dass die Zeigerwerte für andere Anweisungsblöcke, wie beispielsweise TBLK, in diesen Registern gespeichert werden können. In diesem Beispiel verwendet der TBLK-Anweisungsblock das Register 40008 (4x + 7) als Zeiger.
TBLK kopiert Daten von Quelltabellen (siehe nachfolgende Abbildung) in die XMIT-Steuerungstabelle. In diesem Beispiel werden vier Quelltabellen (Block 1 bis Block 4), die jeweils vier Register lang sind, in die XMIT-Steuerungstabelle kopiert (Zielblock). Diese ist vier Register lang (4x + 8 bis 4x + 11). Der Inhalt der Quelltabellen (Block 1 bis Block 4) und die Beschreibung der XMIT-Steuerungstabelle sind in der nachfolgenden Tabelle dargestellt.

Siehe Abbildung zum TBLK-Betrieb.



Siehe Tabelle mit den Inhalten der Quelltabelle und der XMIT-Steuerungstabelle.

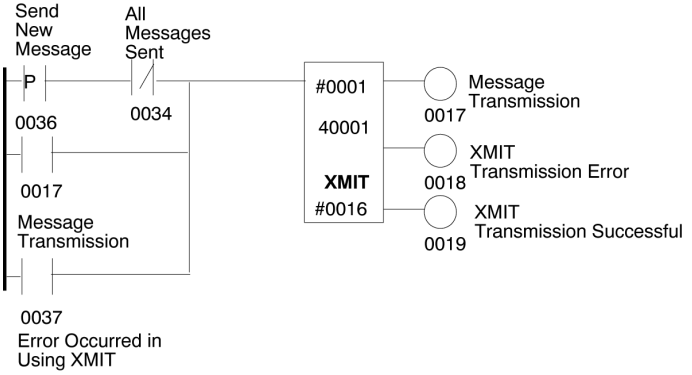
Quelltabellen	Block 1	40200	00000001–00000000 (256 Dec)
		40201	100
		40202	5
		40203	3000
	Block 2	40204	00000001–00000000 (256 Dec)
		40205	105
		40206	5
		40207	3000
	Block 3	40208	00000001 – 00000000 (256 Dec)
		40209	110
		40210	5
		40211	3000
	Block 4	40212	00000001 – 00000000 (256 Dec)
		40213	115
		40214	5
		40215	3000
XMIT-Steuerungstabelle	4x + 8	40009	Befehlswort
	4x + 9	40010	Zeiger auf Meldungstabelle
	4x + 10	40011	Nachrichtenlänge
	4x + 11	40012	Antwort-Timeout (ms)

Wenn Spule 00035 zum ersten Mal auf ON geschaltet wird, kopiert TBLK den Inhalt der ersten Quelltabelle (Block 1 oder 40200 bis 40203) in die Steuerungstabelle (40009 bis 40012). Wenn der Vorgang erfolgreich abgeschlossen wird, wird die nächste Quelltabelle kopiert. So kopiert TBLK die zweite Quelltabelle (Block 2 oder 40204 bis 40207) in die XMIT-Steuerungstabelle (40009 bis 40012). TBLK setzt den Vorgang fort, bis alle vier Modbus-Befehle gesendet wurden (Block 1 bis Block 4). Durch die SUB-Anweisung wird sichergestellt, dass die Tabellenübertragung vollständig abgeschlossen ist. Dabei wird jede Blockübertragung überprüft. Durch die XOR-Anweisung werden alle Register im Bereich (40009 bis 40012) gelöscht.

Verwendung der Ladder Logic für mehrere Modbus-Befehle – Netzwerk Nr. 3

Netzwerk Nr. 3

Netzwerk Nr. 3 sendet die Modbus-Meldung von der Master-SPS an die Slave-SPS. Siehe Abbildung mit den mit Hilfe von XMIT gesendeten Modbus-Befehlen in Netzwerk Nr. 3.



In Netzwerk Nr. 3 wird die Modbus-Meldung mit Hilfe der XMIT-Anweisung gebildet, damit sie von der Master-SPS an die Slave-SPS gesendet werden kann. Der obere Eingang der XMIT-Anweisung bleibt auf ON geschaltet, bis die Modbus-Meldung erfolgreich gesendet wurde. Die XMIT-Steuerungstabelle umfasst 16 Register. In diesem Beispiel beginnt die XMIT-Steuerungstabelle mit Register 40001 und endet mit Register 40016. Die Inhalte dieser Register finden Sie in der nachstehenden Tabelle.

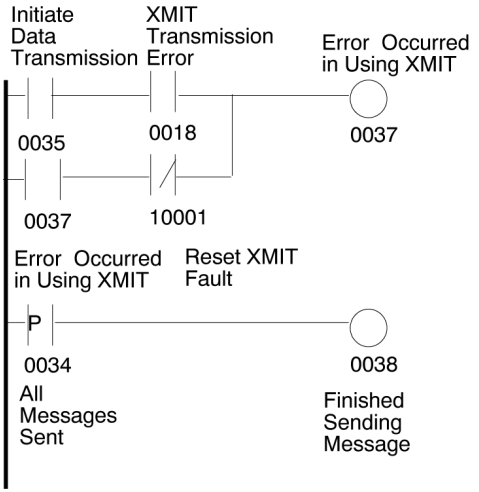
Siehe XMIT-Steuerungstabelle.

Beschreibung	Register	Wert
XMIT-Versionsnummer	40001	201 (oder aktuelle Version)
Fehlerstatus	40002	0
Verfügbar für Benutzer	40003	0 (Kann als Zeiger für Anweisungen, wie beispielsweise TBLK, verwendet werden)
Datenübertragungsrate	40004	9600
Datenbits	40005	8
Parität	40006	0
Stoppbits	40007	1
Verfügbar für Benutzer	40008	0 (Kann als Zeiger für Anweisungen, wie beispielsweise TBLK, verwendet werden)
Befehlswort	40009	0000-0001-0000-0000 (256 Dec)
Zeiger auf Meldungstabelle	40010	100
Nachrichtenlänge	40011	5
Antwort-Timeout (ms):	40012	3000
Wiederholungsgrenze	40013	3
Beginn der Übertragungsverzögerung (ms)	40014	0
Ende der Übertragungsverzögerung (ms)	40015	0
Aktuelle Wiederholung	40016	0

Verwendung der Ladder Logic für mehrere Modbus-Befehle – Netzwerk Nr. 4

Netzwerk Nr. 4

Netzwerk Nr. 4 setzt die XMIT-Anweisung zurück, wenn ein Fehler auftritt. Siehe Abbildung mit den in Netzwerk Nr. 4 zurückgesetzten XMIT-Fehlern.



In Netzwerk Nr. 4 wird Spule 00037 auf ON geschaltet und bleibt im Status ON, bis eine Zurücksetzung durchgeführt wird. Sie sollten wie gewöhnlich bestimmen, wie Fehler je nach Anwendung gehandhabt werden, und Ihre Anwendung zurücksetzen. Spule 00038 wird auf ON gesetzt, wenn vier Modbus-Befehle erfolgreich an die Slave-SPS gesendet wurden. Zum Zurücksetzen (Löschen des Fehlers) muss der obere Eingang des XMIT-Anweisungsblocks für einen SPS-Programmzyklus auf OFF geschaltet werden.

Abschließende Übertragung mehrerer Modbus-Befehle

Abschluss

Die vier Netzwerke der Ladder Logic in diesem Anwendungsbeispiel zeigen, wie einfach eine einzelne XMIT-Anweisung verwendet werden kann, um mehrere Modbus-Befehle von einer Master-SPS an eine Slave-SPS zu senden. Die Programmierung mehrerer Instanzen der XMIT-Steuerungstabelle in der TBLK-Quelltabelle ist eine Methode, die sich hervorragend eignet, um XMIT für eine neue Meldung einzurichten. Daher empfehlen wir, dass Sie diese Methode für alle zukünftigen Anwendungen zur Implementierung der XMIT-Anweisung verwenden.

5.3 Übertragung des Fehlerworts über Impulswahlmodems an den SPS-Slave

Auf einen Blick

Zweck

In diesem Abschnitt wird die Verwendung von Impulswahlmodems für die Übertragung des Fehlerworts beschrieben.

Inhalt dieses Abschnitts

Dieser Abschnitt enthält die folgenden Themen:

Thema	Seite
Übertragung des Fehlerworts an die Slave-SPS über Impulswahlmodems	163
Modem-Setup	164
Einrichten der Master-SPS	165
Verwendung der Ladder Logic für die Fehlerwortübertragung – Netzwerk Nr. 1	166
Verwendung der Ladder Logic für die Fehlerwortübertragung – Netzwerk Nr. 2	167
Verwendung der Ladder Logic für die Fehlerwortübertragung – Netzwerk Nr. 3	170
Verwendung der Ladder Logic für die Fehlerwortübertragung – Netzwerk Nr. 4	172
Abschließende Übertragung des Fehlerworts	173

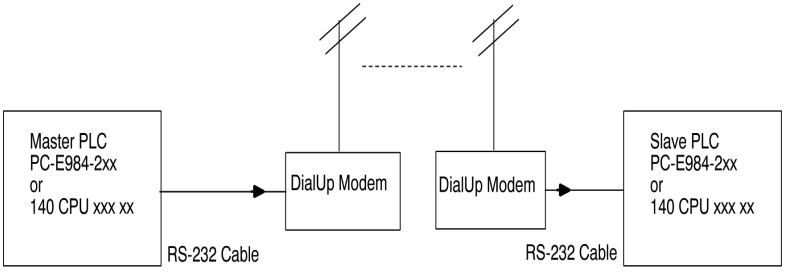
Übertragung des Fehlerworts an die Slave-SPS über Impulswahlmodems

Anwendungsbeispiel für das Senden eines Fehlerworts mit Hilfe eines Impulswahlmodems

Die Ladder Logic in diesem Beispiel zeigt, wie ein einzelnes Fehlerwort (40800) mit Hilfe eines Hayes-kompatiblen Impulswahlmodems an eine Slave-SPS gesendet wird. Diese Logik oder eine Variation dieser Logik kann für Anwendungen verwendet werden, für die ein Bericht für den Umgang mit Ausnahmefehlern erforderlich ist. Wenn ein Fehler auftritt, verwendet die Master-SPS XMIT, um eine Wählzeichenfolge an das Modem zu senden. Wenn eine Verbindung zwischen dem lokalen Modem und dem Modem der Gegenstelle aufgebaut wird, verwendet die Master-SPS XMIT, um eine Modbus-Meldung an die Slave-SPS zu senden. Die Modbus-Meldung schreibt den Inhalt des Fehlerregisters (40800) auf der Master-SPS in das Register (40001) der Slave-SPS. Wenn die Master-SPS eine gültige Antwort von der Slave-SPS erhält, verwendet die Master-SPS XMIT, um eine Auflegezeichenfolge an das lokale Modem zu senden. Somit werden drei Meldungen von der Master-SPS übertragen: Wählnachricht, Modbus-Befehl und Auflegenachricht.

Hardwarekonfiguration

Siehe Abbildung zur Hardwarekonfiguration zur Fehlerwortübertragung.



Hinweis: Diese Anwendung funktioniert ausschließlich mit Impulswahlmodems.

Modem-Setup

Modem-Setup

Si müssen zunächst Ihr Impulswahlmodem initialisieren, um einen ordnungsgemäßen Betrieb mit der XMIT-Anweisung zu gewährleisten. Programmieren Sie eine Initialisierungsmeldung oder ein Kommunikationsprogramm in der Master-SPS, und senden Sie es über die XMIT-Funktion an das Modem. Wir empfehlen zur Initialisierung des Modems die Verwendung eines Terminalprogramms, das die Ladder Logic vereinfacht. In diesem Beispiel wurde ein Kommunikationsprogramm mit dem Namen "Procomm" von DataStrom verwendet, um das Modem zu initialisieren. Wenn möglich, initialisieren Sie alle Impulswahlmodems im System mit Hilfe derselben Initialisierungsmeldung. Die tatsächliche Initialisierungsmeldung und eine Definition der jeweiligen Parameter sind der nachfolgenden Tabelle zu entnehmen.

Siehe Tabelle für Initialisierungsnachricht des Impulswahlmodems.

Initialisierungsnachricht =	AT&F&K0&D0&Q0Q0V1X0
AT=	Vorsicht ***
&F=	Herstellerkonfiguration als aktive Konfiguration wiederherstellen **
&K0=	Lokale Datenflusskontrolle deaktivieren **
&D0=	Status des DTR-Signals ignorieren **
&Q0=	Kommunikation im asynchronen Modus **
Q0=	Ergebniscodes ausgeben *
V1=	Ergebniscodes als Worte anzeigen *
X4=	Einfache Ergebniscodes für Fortschritt des Wählvorgangs anzeigen: Connect, No Carrier und Ring *
E1	Im Befehlsmodus über die Tastatur eingegebene Zeichen auf dem Bildschirm anzeigen *
<CR>	Wagenrücklauf ***
<LF>	Zeilenvorschub ***
* Diese Parameter müssen immer Teil der Initialisierungs-Zeichenkette sein, damit XMIT korrekt arbeitet.	
** Diese Parameter sollten Teil der Initialisierungs-Zeichenkette sein, damit XMIT Nachrichten korrekt zum Modem der Gegenstelle übertragen kann. Nur erfahrene Modem-Benutzer sollten diese Parameter ändern bzw. nicht verwenden.	
*** Diese Parameter werden durch XMIT automatisch hinzugefügt, AT davor und <CR>, <LF> nach der von Ihnen programmierten Meldung.	

Hinweis: Obgleich manche Modem-Hersteller vollständige Hayes-Kompatibilität angeben, können sich die Befehle hier leicht unterscheiden. Daher empfehlen wir, nur die Befehle zu verwenden, deren Definition der oben aufgeführten gleicht.

Einrichten der Master-SPS

Setup der Master-SPS

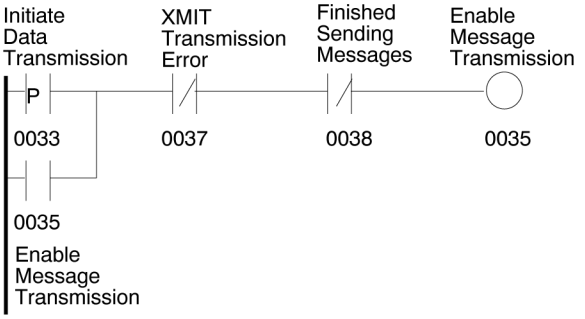
XMIT muss drei Meldungen von der Schnittstelle Nr. 1 der Master-SPS an die Slave SPS übertragen: zwei Modem-Meldungen (Wählen und Auflegen) und eine Modbus-Meldung. Sie müssen diese Meldungen in den Halteregistern der Master-SPS programmieren. Die tatsächlichen Meldungen und deren Inhalt sind der nachfolgenden Tabelle zu entnehmen.
Siehe Tabelle mit den Modem-Meldungen.

Wählen Nachricht	40150=	68ASC	
	40151=	00ASC	
	40152=	32ASC	
	40153=	6 ASC	
Auflegenachricht	40170=	H0ASC	
Modbus-Meldung	40100=	16	Modbus-Funktionscode
	40101=	1	Menge
	40102=	3	Slave-SPS-Adresse
	40103=	1	Slave-SPS-Datenbereich
	40104=	800	Master-SPS-Datenbereich
HINWEIS: Die ATDT-Kopfzeile und die CR/LF-Beisätze werden automatisch gesendet und NICHT in die Länge des Registers zur Meldungssteuerung (4x+10) aufgenommen.			

Verwendung der Ladder Logic für die Fehlerwortübertragung – Netzwerk Nr. 1

Ladder Logic

Netzwerk Nr. 1 sendet die Modbus-Befehle an die Slave-SPS. Die Verweise auf Halteregister, Spulen und Eingänge können entsprechend Ihrer Anwendung geändert werden.
Siehe Abbildung mit den an die Slave-SPS gesendeten Modbus-Befehlen für Netzwerk Nr. 1.

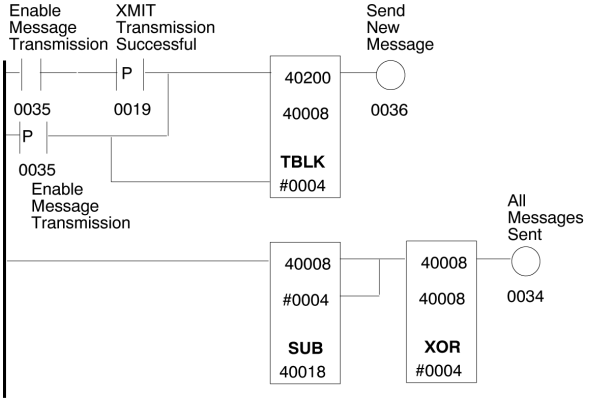


Netzwerk Nr. 1 sendet die Modbus-Befehle an die Slave-SPS, wenn Spule 00033 auf ON geschaltet wird. Spule 00035 bleibt auf ON geschaltet, bis alle drei Meldungen (Modem und Modbus) an die Slave-SPS gesendet wurden. Wenn während einer Modbus-Übertragung an die Slave-SPS ein XMIT-Fehler auftritt, wird Spule 00035 entriegelt.

Verwendung der Ladder Logic für die Fehlerwortübertragung – Netzwerk Nr. 2

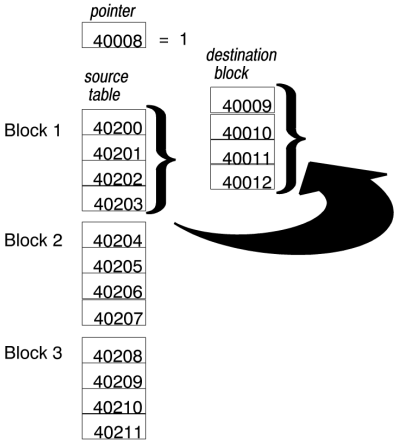
Ladder Logic

Netzwerk Nr. 2 stellt die XMIT-Steuerungstabelle (40001 bis 40016) für eine neue Meldung ein.
Siehe Abbildung zur Einrichtung der XMIT-Steuerungstabelle für Netzwerk Nr. 2.



Zwei Register (4x + 2 und 4x + 7) innerhalb der XMIT-Steuerungstabelle (Registerlänge 16) sind als "Verfügbar für Benutzer" festgelegt, so dass die Zeigerwerte für andere Anweisungsblöcke, wie beispielsweise TBLK, in diesen Registern gespeichert werden können. In diesem Beispiel verwendet der TBLK-Anweisungsblock das Register 40008 (4x + 7) als Zeiger.
TBLK kopiert Daten von Quelltabellen (siehe nachfolgende Abbildung) in die XMIT-Steuerungstabelle. In diesem Beispiel werden drei Quelltabellen (Block 1 bis Block 3), die jeweils vier Register lang sind, in die XMIT-Steuerungstabelle kopiert (Zielblock). Diese ist vier Register lang (4x + 8 bis 4x + 11). Der Inhalt der Quelltabellen (Block 1 bis Block 3) und die Beschreibung der XMIT-Steuerungstabelle sind in der nachfolgenden Tabelle dargestellt.

Siehe Abbildung zum TBLK-Betrieb.



Siehe Tabelle mit den Inhalten der Quelltabelle und der XMIT-Steuerungstabelle.

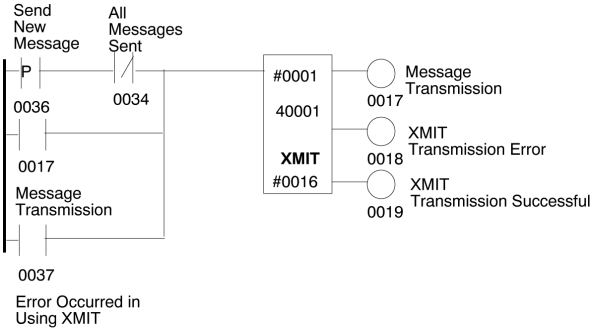
Quelltabellen	Block 1 Wählnachricht: Gesendet an Modem	40200	00000010–00000010 (256 Dec)
		40201	150
		40202	7
		40203	30000
	Block 2 Modbus-Meldung: Gesendet an Slave-SPS	40204	00000001–00000000 (256 Dec)
		40205	100
		40206	5
		40207	3000
	Block 3 Auflegenachricht: Gesendet an Modem	40208	00000010–00000100 (256 Dec)
		40209	170
		40210	2
		40211	30000
XMIT-Steuerungstabelle	4x + 8	40009	Befehlswort
	4x +9	40010	Zeiger auf Meldungstabelle
	4x + 10	40011	Nachrichtenlänge
	4x + 11	40012	Antwort-Timeout (ms):

Block Nr. 1 ist die Wählnachricht, die an das Impulswahlmodem gesendet wird. Das erste Register enthält das Befehlswort. Bit 7 ist auf ON geschaltet und zeigt eine ASCII-Meldung an, und Bit 15 ist auf ON geschaltet und zeigt eine Wählnachricht an. Das zweite Register enthält einen Zeiger für die Wählnachricht, der bei (40150) beginnt. Das dritte Register enthält die Wählnachrichtenlänge (7 Zeichen). Das vierte Register enthält das Timeout für die Wählnachricht (30.000 ms). Wenn ein lokales Modem ein Modem der Gegenstelle anwählt, ist viel Zeit erforderlich, da das lokale Modem einen bestimmten Vorgang durchlaufen muss, um eine Verbindung herzustellen. Daher ist ein Timeout von ca. 3000 ms empfehlenswert. Wenn das Timeout zu kurz ist, gibt XMIT ein Modem-Antwort-Timeout aus.
Block Nr. 2 ist die Modbus-Meldung, die an die Slave-SPS gesendet wird. Das erste Register enthält das Befehlswort. Bit 8 ist auf ON geschaltet und zeigt eine Modbus-Meldung an. Das zweite Register enthält einen Zeiger für die Modbus-Definitionstabelle, der bei (40100) beginnt. XMIT verwendet die hier gespeicherten Informationen, um eine Modbus-Meldung zu bilden. Das dritte Register enthält die Länge der Modbus-Definitionstabelle (5 Register). Das vierte Register enthält das Timeout für die Antwortmeldung der Slave-SPS (3000 ms). Die Antwortzeit der Slave-SPS kann je nach der speziellen Anwendung, die Sie verwenden, geändert werden.

Verwendung der Ladder Logic für die Fehlerwortübertragung – Netzwerk Nr. 3

Ladder Logic

Block Nr. 3 ist die Auflegenachricht, die an die Slave-SPS gesendet wird. Das erste Register enthält das Befehlswort. Bit 14 ist auf ON geschaltet und zeigt eine Auflegenachricht an. Das zweite Register enthält einen Zeiger für die Auflegenachricht, der bei (40170) beginnt. Das dritte Register enthält die Auflegenachricht, die zwei Zeichen umfasst. Das vierte Register enthält das Timeout für die Auflegenachricht (30.000 ms). Wenn das Timeout nicht lang genug ist, gibt XMIT ein Modem-Antwort-Timeout aus. Die Auflegezeit kann je nach der speziellen Anwendung, die Sie verwenden, geändert werden.
Wenn Spule 00035 zum ersten Mal auf ON geschaltet wird, kopiert TBLK den Inhalt der ersten Quelltabelle (Block 1 oder 40200 bis 40203) in die Steuerungstabelle (40009 bis 40012). Wenn der Vorgang erfolgreich abgeschlossen wird, wird die nächste Quelltabelle kopiert. So kopiert TBLK die zweite Quelltabelle (Block 2 oder 40204 bis 40207) in die XMIT-Steuerungstabelle (40009 bis 40012). TBLK setzt den Vorgang fort, bis alle drei Modbus-Befehle gesendet wurden (Block 1 bis Block 3). Durch die SUB-Anweisung wird sichergestellt, dass die Tabellenübertragung vollständig abgeschlossen ist. Dabei wird jede Blockübertragung überprüft.
Durch die XOR-Anweisung werden alle Register im Bereich (40009 bis 40012) gelöscht.
Netzwerk Nr. 3 sendet die Modbus-Meldung von der Master-SPS an die Slave-SPS. Siehe Abbildung mit den mit Hilfe von XMIT gesendeten Modbus-Befehlen in Netzwerk Nr. 3.



In Netzwerk Nr. 3 wird die Modbus-Meldung mit Hilfe der XMIT-Anweisung gebildet, damit sie von der Master-SPS an die Slave-SPS gesendet werden kann. Der obere Eingang der XMIT-Anweisung bleibt auf ON geschaltet, bis die Modbus-Meldung erfolgreich gesendet wurde. Die XMIT-Steuerungstabelle umfasst 16 Register. In diesem Beispiel beginnt die XMIT-Steuerungstabelle mit Register 40001 und endet mit Register 40016. Die Inhalte dieser Register finden Sie in der nachstehenden Tabelle.

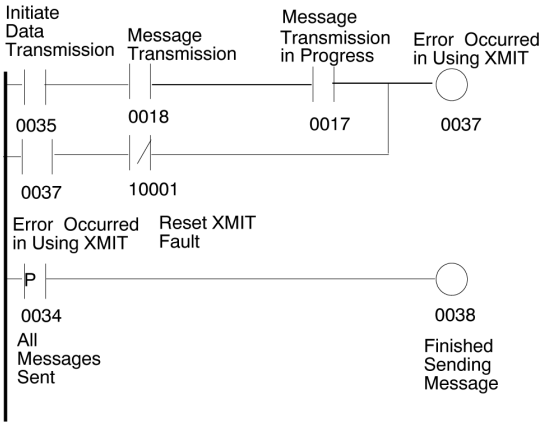
Siehe XMIT-Steuerungstabelle.

Beschreibung	Register	Wert
XMIT-Versionsnummer	40001	201 (oder aktuelle Version)
Fehlerstatus	40002	0
Verfügbar für Benutzer	40003	0 (Kann als Zeiger für Anweisungen, wie beispielsweise TBLK, verwendet werden)
Datenübertragungsrate	40004	9600
Datenbits	40005	8
Parität	40006	0
Stoppbits	40007	1
Verfügbar für Benutzer	40008	0 (Kann als Zeiger für Anweisungen, wie beispielsweise TBLK, verwendet werden)
Befehlswort	40009	0000-0010-0000-0010 (514 Dec)
Zeiger auf Meldungstabelle	40010	150
Nachrichtenlänge	40011	7
Antwort-Timeout (ms):	40012	3000
Wiederholungsgrenze	40013	3
Beginn der Übertragungsverzögerung (ms)	40014	0
Ende der Übertragungsverzögerung (ms)	40015	0
Aktuelle Wiederholung	40016	0

Verwendung der Ladder Logic für die Fehlerwortübertragung – Netzwerk Nr. 4

Ladder Logic

Netzwerk Nr. 4 setzt die XMIT-Anweisung zurück, wenn ein Fehler auftritt. Siehe Abbildung mit den in Netzwerk Nr. 4 zurückgesetzten XMIT-Fehlern.



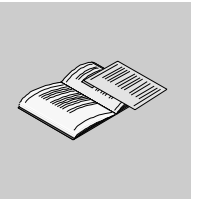
In Netzwerk Nr. 4 wird Spule 00037 auf ON geschaltet und bleibt auf ON geschaltet, bis eine Zurücksetzung durchgeführt wird. Sie sollten wie gewöhnlich bestimmen, wie Fehler je nach Anwendung gehandhabt werden, und Ihre Anwendung zurücksetzen. Spule 00038 wird auf ON geschaltet, wenn alle drei Meldungen (Modem und Modbus) erfolgreich an die Slave-SPS gesendet wurden. Zum Zurücksetzen (Löschen des Fehlers) muss der obere Eingang der XMIT-Anweisungsblocks für einen SPS-Programmzyklus auf OFF geschaltet werden.

Abschließende Übertragung des Fehlerworts

Abschluss

Die vier Netzwerke der Ladder Logic in diesem Anwendungsbeispiel zeigen, wie einfach eine XMIT-Anweisung für die Kommunikation zwischen einer SPS und einem Modem verwendet werden kann. Programmierte ASCII-Meldungen, die in der Master-SPS gespeichert sind, erteilen dem Modem den Befehl zu wählen oder aufzulegen. XMIT sendet die Meldung unter Verwendung der Befehle, die Sie in der Steuerungstabelle programmiert haben, und wartet auf eine Antwort. Die Programmierung mehrerer Instanzen der XMIT-Steuerungstabelle in der TBLK-Quelltabelle ist eine Methode, die sich hervorragend eignet, um XMIT für eine neue Meldung einzurichten. Daher empfehlen wir, dass Sie diese Methode für alle zukünftigen Anwendungen zur Implementierung der XMIT-Anweisung verwenden. Denken Sie daran, dass bestimmte Parameter Bestandteil der Zeichenfolge zur Modem-Initialisierung sein müssen, damit XMIT eine Meldung ordnungsgemäß an ein Modem der Gegenstelle übertragen kann.

Anhang



Auf einen Blick

Inhalt der Anhänge

In den Anhängen erhalten Sie detaillierte technische Informationen zum XMIT-Funktionsblock.

Inhalt dieses Anhangs

Dieser Anhang enthält die folgenden Kapitel:

Kapitel	Kapitelname	Seite
A	Technische Angaben zu XMIT	177

Technische Angaben zu XMIT

A

Auf einen Blick

Einführung

In diesen Unterlagen erhalten Sie detaillierte technische Informationen zum XMIT-Funktionsblock.

Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Abschnitte:

Abschnitt	Thema	Seite
A.1	Arbeiten mit Anfrage-/Antwortparametern	178
A.2	Kabelbelegung und Adapter	181
A.3	Konfiguration von XMIT ausschließlich mit Hayes-kompatiblen Impulswahlmodems	197

A.1 Arbeiten mit Anfrage-/Antwortparametern

Grenzwerte für Modbus-Anfrage-/Antwortparameter

Überblick

In diesem Abschnitt werden die Grenzwerte der Modbus-Anfrage-/Antwortparameter für die folgenden Produktreihen von Schneider Electric Product beschrieben:

- 884, Quantum, Compact, Momentum und Micro
- 584 und 984
- 484
- 184 und 384
- M84

Tabelle der maximalen Parameter für das 884/Quantum.

Funktionscode	Beschreibung	Anfrage	Antwort
1	Spulenstatus lesen	2000 Spulen	2000 Spulen
2	Eingangsstatus lesen	2000-Eingänge	2000-Eingänge
3	Halteregister lesen	125-Register	125-Register
4	Eingangsregister lesen	125-Register	125-Register
5	Einzelne Spule erzwingen	1 Spule	1 Spule
6	Einzelnes Register erzwingen	1 Register	1 Register
15	Mehrere Spulen erzwingen	800 Spulen	800 Spulen
16	Mehrere Register erzwingen	100-Register	100-Register
20	Allgemeine Referenzen lesen	NICHT unterstützt	NICHT unterstützt
21	Allgemeine Referenzen schreiben	NICHT unterstützt	NICHT unterstützt

Tabelle der maximalen Parameter für das 584/984

Funktions-code	Beschreibung	Anfrage	Antwort
1	Spulenstatus lesen	2000 Spulen	2000 Spulen
2	Eingangsstatus lesen	2000-Eingänge	2000-Eingänge
3	Halteregister lesen	125-Register	125-Register
4	Eingangsregister lesen	125-Register	125-Register
5	Einzelne Spule erzwingen	1 Spule	1 Spule
6	Einzelnes Register erzwingen	1 Register	1 Register
15	Mehrere Spulen erzwingen	800 Spulen	800 Spulen
16	Mehrere Register erzwingen	100-Register	100-Register
20	Allgemeine Referenzen lesen (6x)	Die maximale Länge der gesamten Meldung darf 256 Byte NICHT überschreiten	Die maximale Länge der gesamten Meldung darf 256 Byte NICHT überschreiten
21	Allgemeine Referenzen schreiben (6x)	Die maximale Länge der gesamten Meldung darf 256 Byte NICHT überschreiten	Die maximale Länge der gesamten Meldung darf 256 Byte NICHT überschreiten

Tabelle der maximalen Parameter für das 484

Funktionscode	Beschreibung	Anfrage	Antwort
1	Spulenstatus lesen	512 Spulen	512 Spulen
2	Eingangsstatus lesen	512-Eingänge	512-Eingänge
3	Halteregister lesen	254-Register	254-Register
4	Eingangsregister lesen	32-Register	32-Register
5	Einzelne Spule erzwingen	1 Spule	1 Spule
6	Einzelnes Register erzwingen	1 Register	1 Register
15	Mehrere Spulen erzwingen	800 Spulen	800 Spulen
16	Mehrere Register erzwingen	60-Register	60-Register
20	Allgemeine Referenzen lesen	NICHT unterstützt	NICHT unterstützt
21	Allgemeine Referenzen schreiben	NICHT unterstützt	NICHT unterstützt

Tabelle der maximalen Parameter für das 184/384

Funktionscode	Beschreibung	Anfrage	Antwort
1	Spulenstatus lesen	800 Spulen	800 Spulen
2	Eingangsstatus lesen	800-Eingänge	800-Eingänge
3	Halteregister lesen	100-Register	100-Register
4	Eingangsregister lesen	100-Register	100-Register
5	Einzelne Spule erzwingen	1 Spule	1 Spule
6	Einzelnes Register erzwingen	1 Register	1 Register
15	Mehrere Spulen erzwingen	800 Spulen	800 Spulen
16	Mehrere Register erzwingen	100-Register	100-Register
20	Allgemeine Referenzen lesen	NICHT unterstützt	NICHT unterstützt
21	Allgemeine Referenzen schreiben	NICHT unterstützt	NICHT unterstützt

Tabelle der maximalen Parameter für das M84

Funktionscode	Beschreibung	Anfrage	Antwort
1	Spulenstatus lesen	64 Spulen	64 Spulen
2	Eingangsstatus lesen	64-Eingänge	64-Eingänge
3	Halteregister lesen	32-Register	32-Register
4	Eingangsregister lesen	4-Register	4-Register
5	Einzelne Spule erzwingen	1 Spule	1 Spule
6	Einzelnes Register erzwingen	1 Register	1 Register
15	Mehrere Spulen erzwingen	64 Spulen	64 Spulen
16	Mehrere Register erzwingen	32-Register	32-Register

A.2 Kabelbelegung und Adapter

Auf einen Blick

Zweck

In diesem Abschnitt werden zwölf Verkabelungen zur Verbindung von Anschlüssen beschrieben.

- 9-Pin auf 25-Pin
- 9-Pin auf 9-Pin
- RJ-45 (8x8) auf 25-Pin
- RJ-45 (8x8) auf 9-Pin
- RJ-45 (8x8) auf RJ-45 (8x8)

Die Art der Verkabelung ist davon abhängig, ob eine Verbindung zu einem Modem oder zu einem Nullmodem hergestellt wird.

Inhalt dieses Abschnitts

Dieser Abschnitt enthält die folgenden Themen:

Thema	Seite
Kabelbelegungen	182
9-Pin auf 25-Pin (Modem) OHNE RTS/CTS-Datenflusssteuerung	184
9-Pin (RS-232) auf 25-Pin (Modem) mit RTS/CTS-Datenflusssteuerung	185
9-Pin auf 25-Pin (Nullmodem)	186
9-Pin auf 9-Pin (Modem)	187
9-Pin auf 9-Pin (Nullmodem)	188
RJ-45-(8x8) auf 25-poligen Stecker (Modem) (Konfiguration A)	189
RJ-45-(8x8) auf 25-poligen Stecker (Modem) (Konfiguration B)	190
RJ-45-(8x8) auf 25-Pin-Stecker (Nullmodem)	191
RJ-45-(8x8) 9-Pin-Stecker (Modem) (Konfiguration A)	192
RJ-45-(8x8) 9-Pin-Stecker (Modem) (Konfiguration B)	193
RJ-45 (8x8) 9-poliger Stecker (Nullmodem)	194
RJ-45-(8x8) auf RJ-45-(8x8) (Modem)	195
Teilesätze für Kabeladapter	196

Kabelbelegungen

Überblick

Sechs der folgenden Kabelbelegungskombinationen sind als Adapterteilesätze verfügbar.

Beschreibung	Beschriebene Kabelbelegung
1	9-Pin auf 25-Pin (Modem) OHNE RTS/CTS-Datenflusssteuerung
2	9-Pin (RS-232) auf 25-Pin (Modem) mit RTS/CTS-Datenflusssteuerung
3	9-Pin auf 25-Pin (Nullmodem)
4	9-Pin auf 9-Pin (Modem)
5	9-Pin auf 9-Pin (Nullmodem)
6	RJ-45-(8x8) auf 25-Pin-Stecker (Modem): Adapterteilesatz: 110XCA20401
7	RJ-45-(8x8) auf 25-Pin-Stecker (Modem): Adapterteilesatz: 110XCA20401
8	RJ-45-(8x8) auf 25-Pin-Stecker (Nullmodem): Adapterteilesatz: 110XCA20401
9	RJ-45-(8x8) auf 9-Pin-Stecker (Modem): Adapterteilesatz: 110XCA20301
10	RJ-45-(8x8) auf 9-Pin-Stecker (Modem): Adapterteilesatz: 110XCA20301
11	RJ-45-(8x8) auf 9-Pin-Stecker (Nullmodem): Adapterteilesatz: 110XCA20301
12	RJ-45-(8x8) auf RJ-45-(8x8) (Modem)

Kabelbelegungen der Schnittstellen

Fertigen Sie ein Schnittstellenkabel zwischen Ihrer SPS und dem Modem oder Drucker an.

Wählen Sie eine der beiden Optionen für die Kabelverbindung aus. Verbinden Sie das Kabel

- mit Schnittstelle Nr. 1 der SPS und mit der RS-232-Schnittstelle (9-Pin-Stecker) der Modem- oder Druckerschnittstelle (25-Pin-Stecker)
- direkt mit der Modbus-Schnittstelle einer anderen SPS.

Da der XMIT-Funktionsblock viele Modems und Drucker unterstützt, können sich die Kabelbelegungen unterscheiden.

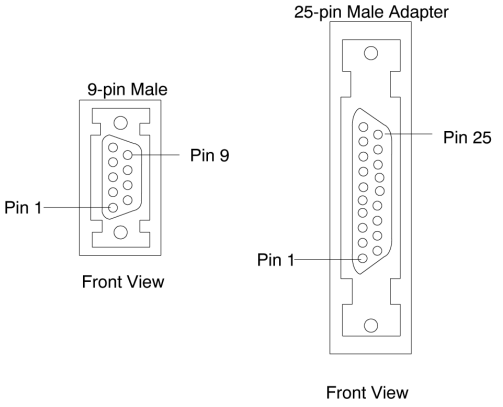
Hinweis: Das Dokument "Modicon 309COM4550x XMIT-Loadable – Readme" (GI-XMIT-RMF) enthält eine Liste mit Angaben zur Kabelbelegung für Geräte, die für die Schnittstelle Nr. 1 der Modbus-Master-SPS getestet wurden.

Hinweis: Im Jahr 1999 wurde die Bezeichnung RS-232 in EIA-232 geändert.

9-Pin auf 25-Pin (Modem) OHNE RTS/CTS-Datenflusssteuerung

9-Pin auf 25-Pin (Modem) OHNE RTS/CTS-Datenflusssteuerung

Steckverbinder: Vorderansichten



Verkabelung für 9-Pin auf 25-Pin (Modem) OHNE RTS/CTS-Datenflusssteuerung

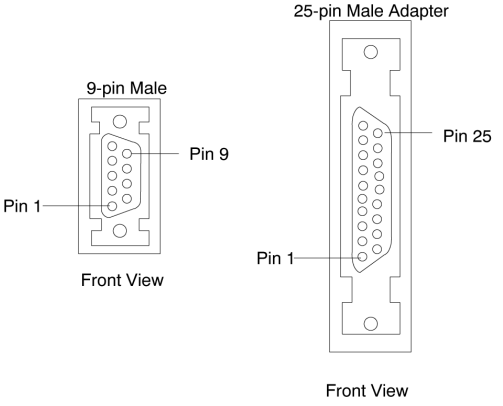
Steckerbelegungen

9-Pin-Stecker			25-Pin D-Gehäuse	
RXD	2	⟷	3	RXD
TXD	3	⟷	2	TXD
RTS	7	⟷	4	RTS
CTS	8	⟷	5	CTS
DSR	4	⟷	6	DSR
DTR	6	⟷	20	DTR
Masse	5	⟷	7	Masse

9-Pin (RS-232) auf 25-Pin (Modem) mit RTS/CTS-Datenflusssteuerung

9-Pin (RS-232) auf 25-Pin (Modem) mit RTS/CTS-Daten-flusssteuerung

-Steckverbindern: Vorderansichten



Verkabelung für 9-Pin auf 25-Pin (Modem) mit RTS/CTS-Datenflusssteuerung

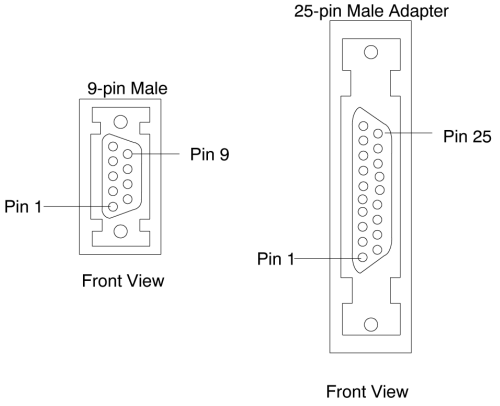
Steckerbelegungen

9-Pin-Stecker			25-Pin D-Gehäuse	
RXD	2	⟷	3	RXD
TXD	3	⟷	2	TXD
RTS	7	⟷	4	RTS
CTS	8	⟷	5	CTS
DSR	4	⟷	6	DSR
DTR	6	⟷	20	DTR
Masse	5	⟷	7	Masse

9-Pin auf 25-Pin (Nullmodem)

9-Pin auf 25-Pin (Nullmodem)

Steckverbinder: Vorderansichten



Verkabelung für 9-Pin auf 25-Pin (Nullmodem)

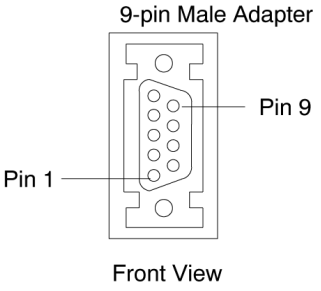
Steckerbelegungen

9-Pin-Stecker			25-Pin D-Gehäuse	
RXD	2	◄──────►	2	TXD
TXD	3	◄──────►	3	RXD
RTS	7	◄──┐ ┌──►	4	RTS
CTS	8	◄──┘ └──►	5	CTS
DSR	4	◄──┐ ┌──►	6	DSR
DTR	6	◄──┘ └──►	20	DTR
Masse	5	◄──────►	7	Masse

9-Pin auf 9-Pin (Modem)

9-Pin auf 9-Pin (Modem)

Steckverbinder: Vorderansicht



Verkabelung für 9-Pin auf 9-Pin (Modem)

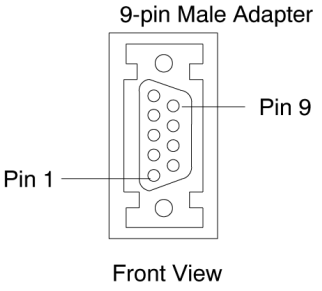
Steckerbelegungen

9-Pin-Stecker			9-Pin-Stecker	
RXD	2	◄——►	2	TXD
TXD	3	◄——►	3	RXD
RTS	7	◄——►	7	RTS
CTS	8	◄——►	8	CTS
DSR	4	◄—┐ ┌—	4	DSR
DTR	6	◄—┘ ┌—►	6	DTR
Masse	5	◄——►	5	Masse

9-Pin auf 9-Pin (Nullmodem)

9-Pin auf 9-Pin (Nullmodem)

Steckverbinder: Vorderansicht



Verkabelung für 9-Pin auf 9-Pin (Nullmodem)

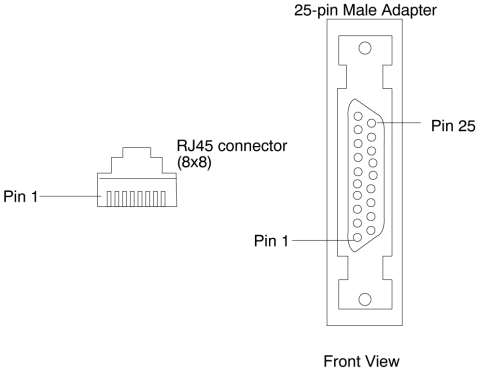
Steckerbelegungen

9-Pin-Stecker			9-Pin-Stecker	
RXD	2	◄──────►	3	RXD
TXD	3	◄──────►	2	TXD
RTS	7	◄──┐ ┌──►	7	RTS
CTS	8	◄──┘ └──►	8	CTS
DSR	4	◄──┐ ┌──►	4	DSR
DTR	6	◄──┘ └──►	6	DTR
Masse	5	◄──────►	5	Masse

RJ-45-(8x8) auf 25-poligen Stecker (Modem) (Konfiguration A)

RJ-45-(8x8) auf 25-poligen Stecker (Modem) 110XCA20401

Steckverbinder: Vorderansichten



Verkabelung für den RJ-45-(8x8) auf 25-poligen Stecker (Modem); Teilesatz für Adapter: 110XCA20401

Steckerbelegungen

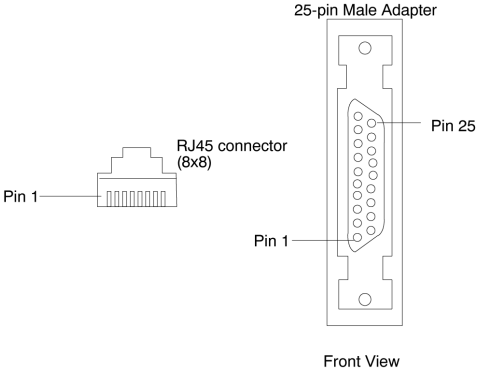
RJ-45-Stecker			25-Pin D-Gehäuse	
RXD	4		3	RXD
TXD	3		2	TXD
RTS	6		4	RTS
CTS	7		5	CTS
Masse	5		7	Masse
DSR	2		6	DSR
			20	DTR
Gehäuse-Masse	8		1	Gehäuse-Masse

Hinweis: Pin 1 des RJ-45 erhält 5 V von der SPS.

RJ-45-(8x8) auf 25-poligen Stecker (Modem) (Konfiguration B)

RJ-45-(8x8) auf 25-poligen Stecker (Modem) 110XCA20401

Steckverbinder: Vorderansichten



Verkabelung für den RJ-45-(8x8) auf 25-poligen Stecker (Modem); Teilesatz für Adapter: 110XCA20401

Steckerbelegungen

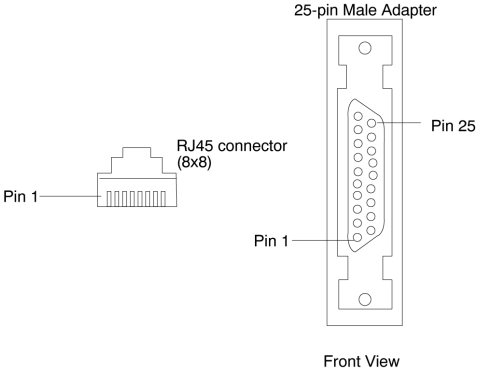
RJ-45-Stecker			25-Pin D-Gehäuse	
RXD	4	◄——►	3	RXD
TXD	3	◄——►	2	TXD
RTS	6	◄——►	4	RTS
CTS	7	◄——►	5	CTS
Masse	5	◄——►	7	Masse
		——►	6	DSR
		◄——	20	DTR
Gehäuse-Masse	8	◄——►	1	Gehäuse-Masse

Hinweis: Pin 1 des RJ-45 erhält 5 V von der SPS.

RJ-45-(8x8) auf 25-Pin-Stecker (Nullmodem)

RJ-45-(8x8) auf 25-Pin-Stecker (Nullmodem) 110XCA20401

Steckverbinder: Vorderansichten



Verkabelung für den RJ-45-(8x8) auf 25-poligen Stecker (Nullmodem); Teilesatz für Adapter: 110XCA20401

Steckerbelegungen

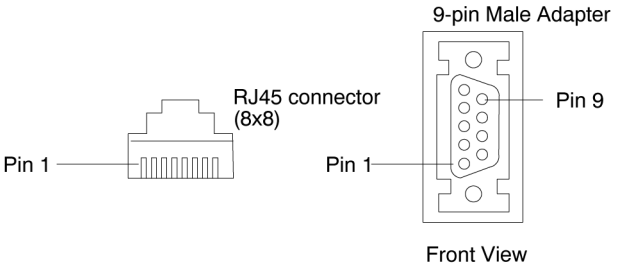
RJ-45-Stecker			25-Pin D-Gehäuse	
RXD	4	⟷	2	TXD
TXD	3	⟷	3	RXD
RTS	6	⟷	4	RTS
CTS	7	⟷	5	CTS
Masse	5	⟷	7	Masse
DSR	2	⟷	6	DSR
			20	DTR
Gehäuse-Masse	8	⟷	1	Gehäuse-Masse

Hinweis: Pin 1 des RJ-45 erhält 5 V von der SPS.

RJ-45-(8x8) 9-Pin-Stecker (Modem) (Konfiguration A)

RJ-45-(8x8) auf 9-poligen Stecker (Modem) 110XCA20301

Steckverbinder: Vorderansichten



Verkabelung für den RJ-45-(8x8) auf 9-poligen Stecker (Modem); Teilesatz für Adapter: 110XCA20301

Steckerbelegungen

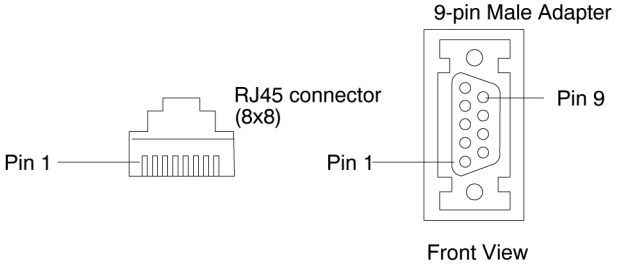
RJ-45-Stecker			9-Pin-Stecker	
RXD	4	→	2	RXD
TXD	3	→	3	TXD
RTS	6	→	7	RTS
CTS	7	←	8	CTS
Masse	5	→	5	Masse
DSR	2	→	6	DSR
			4	DTR
Gehäuse-Masse	8	→		Stecker-gehäuse

Hinweis: Pin 1 des RJ-45 erhält 5 V von der SPS.

RJ-45-(8x8) 9-Pin-Stecker (Modem) (Konfiguration B)

RJ-45-(8x8) auf 9-poligen Stecker (Modem) 110XCA20301

Steckverbinder: Vorderansichten



Verkabelung für den RJ-45-(8x8) auf 9-poligen Stecker (Modem); Teilesatz für Adapter: 110XCA20301

Steckerbelegungen

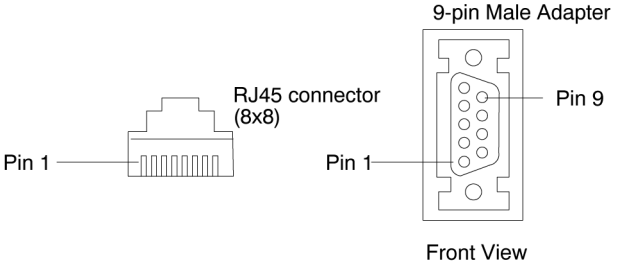
RJ-45-Stecker			9-Pin-Stecker	
RXD	4	⟷	2	RXD
TXD	3	⟷	3	TXD
RTS	6	⟷	7	RTS
CTS	7	⟷	8	CTS
Masse	5	⟷	5	Masse
		⟷	6	DSR
		⟷	4	DTR
Gehäuse-Masse	8	⟷		Stecker-gehäuse

Hinweis: Pin 1 des RJ-45 erhält 5 V von der SPS.

RJ-45 (8x8) 9-poliger Stecker (Nullmodem)

RJ-45-(8x8) auf 9-poligen Stecker (Nullmodem) 110XCA20301

Steckverbinder: Vorderansichten



Verkabelung für den RJ-45-(8x8) auf 9-Pin-Stecker (Nullmodem); Teilesatz für Adapter: 110XCA20301

Steckerbelegungen

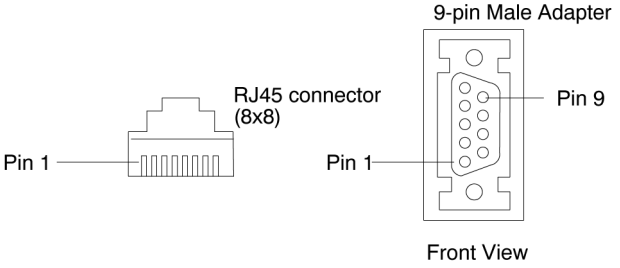
RJ-45-Stecker			9-Pin D-Gehäuse	
RXD	4	⟷	3	TXD
TXD	3	⟷	2	RXD
RTS	6	⟷	7	RTS
CTS	7	⟷	8	CTS
Masse	5	⟷	5	Masse
DSR	2	⟷	4	DTR
			6	DTR
Gehäuse-Masse	8	⟷		Stecker-gehäuse

Hinweis: Pin1 des RJ-45 erhält 5 V von der SPS.

RJ-45-(8x8) auf RJ-45-(8x8) (Modem)

RJ-45-(8x8) auf RJ-45-(8x8) (Modem)

Steckverbinder: Vorderansichten



Verkabelung für RJ-45-(8x8) auf RJ-45-(8x8) (Modem)

Steckerbelegungen

RJ-45-Stecker			RJ-45-Stecker	
RXD	4	◄——►	4	RXD
TXD	3	◄——►	3	TXD
RTS	6	◄——►	6	RTS
CTS	7	◄——►	7	CTS
Masse	5	◄——►	5	Masse
DSR	2	◄——►	2	DSR
Gehäuse-Masse	8	◄——►	1	Gehäuse-Masse

Hinweis: Pin 1 des RJ-45 erhält 5 V von der SPS.

Teilesätze für Kabeladapter

Verfügbare Teilesätze für Kabel

Anstatt Kabeladapter für Ihre RJ-45-Anforderungen selbst anzufertigen, können Sie Teilesätze für Kabeladapter käuflich erwerben.
Teilesätze für Kabeladapter

Beschreibung	Teilenummer
RJ-45 auf 25-Pin (Stecker)	110XCA20401
RJ-45 auf 9-Pin (Stecker)	110XCA20301
RJ-45 auf 9-Pin (Buchse)	110XCA20302
RJ-45 auf 25-Pin (Buchse)	110XCA20402

A.3 Konfiguration von XMIT ausschließlich mit Hayes-kompatiblen Impulswahlmodems

Auf einen Blick

Zweck

In diesem Abschnitt wird die Verwendung des XMIT-Blocks und der drei Befehle beschrieben, die für die Kommunikation über Impulswahlmodems erforderlich sind.

1. Modem initialisieren
2. Modem wählen
3. Modem auflegen

Inhalt dieses Abschnitts

Dieser Abschnitt enthält die folgenden Themen:

Thema	Seite
Verwendung der XMIT-Konfiguration ausschließlich mit Hayes-kompatiblen Impulswahlmodems	198
Verwendung von Initialisierungsnachrichten mit Hayes-Modems	199
Verwendung von Impulswahlmodem-Meldungen mit Hayes-Modems	201
Verwendung von Auflegenachrichten ausschließlich mit Hayes-kompatiblen Impulswahlmodems	202

Verwendung der XMIT-Konfiguration ausschließlich mit Hayes-kompatiblen Impulswahlmodems

Grundlegende Befehle

Es gibt drei Befehle, mit denen Sie vertraut sein müssen, wenn Sie über Wählmodem mit XMIT kommunizieren. Diese Befehle sind:

1. Modem initialisieren
2. Modem wählen
3. Modem auflegen

Bevor eine ASCII-Nachricht oder eine Modbus-Nachricht das Modem passieren kann, müssen Sie zuerst eine Initialisierungs-Zeichenkette und dann eine Wähl-Zeichenkette an das Modem senden. Sobald das Modem die Telefonnummer gewählt und eine Verbindung zum Modem der Gegenstelle hergestellt hat, können Sie beliebig viele ASCII-Nachrichten oder Modbus-Nachrichten über das Modem schicken. Um mehrere Nachrichten zu senden, inkrementieren Sie nach jeder erfolgreichen XMIT-Operation den Nachrichtenzähler auf die nächste Nachricht. Wenn alle Nachrichten versendet sind, können Sie die Auflegezeichenkette an das Modem schicken.

Verwendung von Initialisierungsnachrichten mit Hayes-Modems

Initialisierungs-nachricht

Die Initialisierungsnachricht ist eine gewöhnliche ASCII-Nachricht und kann maximal 512 Zeichen lang sein. In der Regel sind jedoch 50 Zeichen mehr als ausreichend, um ein Modem zu initialisieren. Sie können jeden beliebigen Hayes-AT-Befehl als Teil der Initialisierungs-Zeichenkette verwenden. Wir empfehlen die folgenden Befehle zur Initialisierung eines Modems zur Verwendung mit XMIT. Initialisierungsnachricht für Impulswahlmodem.

Initialisierungsnachricht =	**AT&F&K0&Q0&D0V1Q0X0E1**
AT=	Modem selbst kalibrieren*
&F=	Herstellerkonfiguration als aktive Konfiguration wiederherstellen*
&K0=	Lokale Datenflusskontrolle deaktivieren**
&Q0=	Kommunikation im asynchronen Modus**
&D0=	Status des DTR-Signals ignorieren*
V1=	Ergebniscodes als Wörter anzeigen* Wenn V1 nicht verwendet wird oder falls das Modem nicht in der Lage ist, Antworten in Wortform wiederzugeben, gibt der XMIT-Funktionsbaustein Fehler 117 aus (Modem-Antwort-Timeout).
Q0=	Ergebniscodes ausgeben*
X4=	Einfache Ergebniscodes für Fortschritt des Wählvorgangs anzeigen: Verbinden, kein Carrier und Ring*
E1=	Im Befehlsmodus über die Tastatur eingegebene Zeichen auf dem Bildschirm anzeigen*
*Diese Parameter müssen immer Teil der Initialisierungs-Zeichenkette sein, damit XMIT korrekt arbeitet.	
**Diese Parameter sollten Teil der Initialisierungs-Zeichenkette sein, damit XMIT Nachrichten korrekt zum Modem der Gegenstelle übertragen kann. Nur erfahrene Modem-Benutzer sollten diese Parameter ändern.	

Hinweis: Obgleich manche Modem-Hersteller vollständige Hayes-Kompatibilität angeben, können sich die Befehle hier leicht unterscheiden. Daher empfehlen wir, nur die Befehle zu verwenden, deren Definition der oben aufgeführten gleicht.

Die Initialisierungsnachricht muss immer mit dem Standard-Hayes-Befehl AT beginnen. Der XMIT-Block leitet Modem-Befehlsnachrichten automatisch mit AT ein und hängt die Zeichen Wagenrücklauf (0x0D) und Zeilenvorschub (0x0A) an die Nachricht an, da diese von allen Modem-Steuerungsnachrichten benötigt werden. Andere (nicht steuernde) ASCII-Nachrichten müssen nicht mit Wagenrücklauf und Zeilenvorschub enden.
Beispiel: Eine typische Initialisierungsnachricht, die XMIT an das Modem schickt.

- Meldung = (AT)&F&K0&Q0&D0V1X0Q0 (<CR><LF>)*
- Länge = 17 Zeichen

*Zeichen in Klammern werden automatisch geschickt.
Beispiel: Die Initialisierungsnachricht kann auch zum Setzen von S-Registern des Modems verwendet werden.

- Meldung = (AT)S0=1 (<CR><LF>)*
- Länge = 4 Zeichen

*Zeichen in Klammern werden automatisch geschickt.
Damit XMIT eine Initialisierungsnachricht an das Modem schickt, müssen Bit 7 und 16 des Befehlsworts auf ON gesetzt sein. Wenn Bit 15 auf ON gesetzt ist, dürfen die Bits 15 und 14 nicht auf ON gesetzt sein, weil sonst XMIT die Operation nicht erfolgreich abschließen kann. Damit die Nachricht tatsächlich gesendet wird, muss der obere Eingang von XMIT auf ON gesetzt werden und auf ON bleiben, bis die Operation abgeschlossen ist oder ein Fehler auftritt. Wenn XMIT feststellt, dass die Nachricht erfolgreich an das Modem geschickt wurde, setzt es den unteren Ausgang auf ON. Wenn ein Fehler auftritt, wird der mittlere Ausgang auf ON gesetzt. Der obere Ausgang ist auf ON geschaltet, während die Nachricht an das Modem geschickt wird.

Hinweis: PROGRAMMIERUNG DER KOP-LOGIK REDUZIEREN
Um einen Teil der Programmierung der Ladder Logic zu umgehen, können Sie das Modem über ein Terminalprogramm mit Parametern initialisieren und XMIT nicht verwenden. Sobald die Parameter sich im Modemspeicher befinden, können sie mit einem AT-Befehl im nichtflüchtigen Speicher gespeichert werden, gewöhnlich ist dies der Befehl &W.

Verwendung von Impulswahlmodem-Meldungen mit Hayes-Modems

Wählnachrichten

Die Wählnachricht wird dazu verwendet, eine Telefonnummer an das Modem zu senden. In der Nachricht sollten nur AT-Befehle verwendet werden, die in Bezug zum Wählen einer Nummer stehen.

Beispiel: Telefonnummer mittels Tonwahl wählen.

- Meldung = (ATDT)6800326 (<CR><LF>)*
- Länge = 7 Zeichen

*Zeichen in Klammern werden automatisch geschickt.

Beispiel: Telefonnummer mittels Pulswahl wählen.

- Meldung = (ATDP) 6800326 (<CR><LF>)*
- Länge = 7 Zeichen

*Zeichen in Klammern werden automatisch geschickt.

Beispiel: Telefonnummer mittels Tonwahl wählen, vor dem Wählen der Nummer auf Wählton warten und vor dem Wählen der restlichen Nummer eine Pause machen.

- Meldung = (ATDT)W, 6800326 (<CR><LF>)*
- Länge = 9 Zeichen

*Zeichen in Klammern werden automatisch geschickt.

Damit XMIT eine Tonwahlnachricht an das Modem schickt, müssen Bit 7 und 15 des Befehlsworts auf ON gesetzt sein. Wenn Bit 16 auf ON gesetzt ist, dürfen die Bits 16 und 14 nicht auf ON gesetzt sein, weil sonst XMIT die Operation nicht erfolgreich abschließen kann. Damit die Nachricht tatsächlich gesendet wird, muss der obere Eingang von XMIT auf ON gesetzt werden und auf ON bleiben, bis die Operation abgeschlossen ist oder ein Fehler auftritt. Wenn XMIT feststellt, dass die Nachricht erfolgreich an das Modem geschickt wurde, setzt es den unteren Ausgang auf ON. Wenn ein Fehler auftritt, wird der mittlere Ausgang auf ON gesetzt. Der obere Ausgang ist auf ON geschaltet, während die Nachricht an das Modem geschickt wird.

Hinweis: Einstellung des Timeout-Werts
Da es lange dauert, bis ein lokales Modem eine Verbindung zum Modem der Gegenstelle aufgebaut hat, sollte der Timeout-Wert in Register (4x + 11) so lange wie möglich gewählt werden, wenn eine Wählnachricht an ein Modem geschickt wird. Setzen Sie beispielsweise beim Senden einer Wählnachricht den Timeout auf 30.000 ms. Wenn der Timeout-Wert zu klein ist, gibt XMIT einen Nachrichten-Timeout aus. Unter Umständen müssen Sie verschiedene Einstellungen testen, um die optimale Zeit zu finden.

Verwendung von Auflegenachrichten ausschließlich mit Hayes-kompatiblen Impulswahlmodems

Auflegenachricht

Die Auflegenachricht wird zum Auflegen des Modems verwendet. In dieser Nachricht sollten nur AT-Befehle verwendet werden, die in Bezug zum Auflegen des Modems stehen.

Ein Beispiel für eine typische Auflegenachricht wird nachfolgend gezeigt.

- Meldung = (+++AT) H0 (<CR><LF>)*
- Länge = 2 Zeichen

*Zeichen in Klammern werden automatisch geschickt.

Wenn die Auflegenachricht an ein Modem geschickt wird, das bereits mit einem Modem der Gegenstelle verbunden ist, muss XMIT zuerst das lokale Modem in den Befehlsmodus versetzen, indem die Escape-Sequenz +++ an das Modem gesendet wird. XMIT nimmt an, dass die Zeichenfolge +++ das Modem in den Befehlsmodus versetzt. Einige Modemhersteller lassen den Benutzer diese Standard-Escape-Sequenz verändern. Damit XMIT korrekt funktioniert, sollte das Modem so eingestellt werden, dass es die Escape-Sequenz +++ akzeptiert.

Damit XMIT eine Auflegenachricht an das Modem schickt, müssen Bit 7 und 14 des Befehlsworts auf ON gesetzt sein. Wenn Bit 16 auf ON gesetzt ist, dürfen die Bits 16 und 15 nicht auf ON gesetzt sein, weil sonst XMIT die Operation nicht erfolgreich abschließen kann. Damit die Nachricht tatsächlich gesendet wird, muss der obere Eingang von XMIT auf ON gesetzt werden und auf ON bleiben, bis die Operation abgeschlossen ist oder ein Fehler auftritt. Wenn XMIT feststellt, dass die Nachricht erfolgreich an das Modem geschickt wurde, setzt es den unteren Ausgang auf ON. Wenn ein Fehler auftritt, wird der mittlere Ausgang auf ON gesetzt. Der obere Ausgang ist auf ON geschaltet, während die Nachricht an das Modem geschickt wird.

Hinweis: Einstellung des Timeout-Werts
Da es lange dauert, bis ein lokales Modem auflegt, nachdem es die Auflegenachricht empfangen hat, sollte der Timeout-Wert in Register (4x + 11) sehr lang gewählt werden, wenn eine Wählnachricht an ein Modem geschickt wird. Setzen Sie beispielsweise beim Senden einer Wählnachricht den Timeout auf 30.000 ms. Wenn der Timeout-Wert zu klein ist, gibt XMIT einen Nachrichten-Timeout aus. Unter Umständen müssen Sie verschiedene Einstellungen testen, um die optimale Zeit zu finden.

Index

A

Anwendungsbeispiel
- Impulswahlmodem, 163
- Radio-/Mietleitungs-Modem, 152

B

Befehlswort
- Bit-Definitionen, 83

D

Datenübertragungsraten-Register
- Bereiche, 81

F

Fehlerstatusregister
- Konvertierungsblock, 118
- Schnittstellenstatusblock, 109
- XMIT-Kommunikationsblock, 79

Funktionscodes der Modbus-Definitionstabelle
- 01 bis 06, 15 und 16, 96
- 08, 98

Funktionscodes der Modbus-Funktionsdefinitionstabelle
- 20, 21, 100

G

Gültige Bereiche
- Konvertierungssteuerungstabelle, 116
- Tabelle für die Kommunikationssteuerung, 75
- Tabelle für Schnittstellenstatussteuerung, 108

I

Installation
- Datei DXFDT.SYS, 25
- NSUP.EXE, Datei, 27
- XMIT.EXE, Datei, 29
- XMIT.ZMM, Datei, 25
- XMIT1968.HLP, Datei, 26

K

Kabeladapter, Teilesätze, 182

Kabelbelegung
- 9-Pin auf 25-Pin (Modem) ohne RTS/CTS-Datenflusssteuerung, 184
- 9-Pin auf 9-Pin (Modem), 187
- 9-Pin auf 9-Pin (Nullmodem), 188
- RJ45-(8x8) auf 25-Pin (Modem), 190, 191
- RJ-45-(8x8) auf 25-Pin-Stecker (Modem), 189
- RJ45-(8x8) auf 9-Pin (Modem), 193

RJ45-(8x8) auf 9-Pin-Stecker (Modem), 192
RJ45-(8x8) auf RJ45-(8x8) (Modem), 195
Kabelbelegungen, 182
9-Pin auf 25-Pin (Modem) mit RTS/C, 185
9-Pin auf 25-Pin (Nullmodem), 186
RJ45-(8x8) auf 9-poligen Stecker (Nullmodem), 194

L

Loadables
Ladereihenfolge, 27

O

Opcodes
Konfliktlösung
Modsoft, 28
NSUP-Loadable, 28
XMIT-Loadable, 31

P

Parameter-Grenzwerte
184/384, 180
484, 179
584/984, 179
884/Quantum, 178
Parametergrenzwerte
M84, 180

T

Teilesätze
Für Kabeladapter, 182

U

Übertragung
Datei XMIT.EXE, 23

V

Vermeidung konkurrierender Zugriffe, 13
Vermeidung von Datenkollisionen, 13

X

XMIT-Kommunikation
Fehlercodes, 52
Modbus-Funktionscodes, 50
Steuerungstabelle, 51
XMIT-Konvertierung
Fehlercodes, 56
XMIT-Schnittstellenstatus
Fehlercodes, 54